ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – TE 141599

RANCANG BANGUN SISTEM KEAMANAN LEMARI BENDA BERHARGA BERBASIS MODUL PENGENALAN SUARA

Gusti Paring
NRP 07111645000049

Dosen Pembimbing
Dr. Ir. Hendra Kusuma, M.Eng.Sc
Dr. Ir. Totok Mujiono, M.Kom

PROGRAM STUDI S1 TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT – TE 141559

DESIGN SECURITY SYSTEM SAFE STORAGE BASED VOICE RECOGNITION MODULE

Gusti Paring
NRP 07111645000049

Advisor
Dr. Ir. Hendra Kusuma, M.Eng.Sc
Dr. Ir. Totok Mujiono, M.Kom

ELECTRICAL ENGINEERING S1 STUDY PROGRAM
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

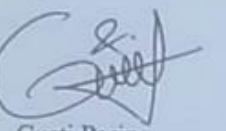
PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**SISTEM KEAMANAN LEMARI BENDA BERHARGA BERBASIS MODUL PENGENALAN SUARA** " adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 02 Juli 2018



Gusti Paring
NRP 07111645000049

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN SISTEM KEAMANAN LEMARI BENDA BERHARGA BERBASIS MODUL PENGENALAN SUARA

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Ir. Hendra Kusuma, M. Eng.Sc	Dr. Ir. Totok Mujiono, M. Kom
NIP. 196409021989031003	NIP. 196504221989031001

SURABAYA
JULI, 2018

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN SISTEM KEAMANAN LEMARI BENDA BERHARGA BERBASIS MODUL PENGENALAN SUARA

Nama : Gusti Paring
Pembimbing I : Dr. Ir. Hendra Kusuma, M. Eng.Sc
Pembimbing II : Dr. Ir. Totok Mujiono, M.Kom

ABSTRAK

Lemari benda berharga berfungsi untuk menyimpan benda yang bersifat penting, dan umumnya masih menggunakan kunci konvensional dimana ada kemungkinan kunci hilang maupun dicuri. Pada Tugas Akhir ini membuat sistem keamanan lemari benda berharga berbasis modul pengenalan suara yang dimana suara digunakan sebagai *password* untuk membuka lemari benda berharga dan masukkan *password* pada *keypad* sebagai pengamanan kedua setelah suara. Alat ini mampu menyimpan tiga suara pengguna yang berbeda, serta dilengkapi dengan fasilitas basis data untuk menyimpan aktifitas penggunaan lemari tersebut yang dikirim pada google sheet. Pemrosesan suara menggunakan STM32F407VG dengan menggunakan modul pengenalan suara V3. Hasil dari tugas akhir ini mampu mengenali suara pengguna lemari benda berharga dengan kata – kata yang berbeda dari tiga pengguna dengan intensitas suara maksimal yang dapat diterima sebesar 94,6 dB dan tingkat intensitas kebisingan tidak melebihi 85,5 dB. Hasil pengenalan suara pada modul yang dapat diproses adalah sebesar 80% dari dua puluh kali percobaan. Serta dapat mencatat waktu pengguna pada basis data secara *Real-Time*.

Kata Kunci : STM32F407VG, Modul Pengenalan Suara, Lemari Benda Berhaga

-----Halaman ini sengaja dikosongkan-----

DESIGN SECURITY SYSTEM SAFE STORAGE BASED VOICE RECOGNITION MODULE

Name : Gusti Paring
Advisor 1st : Dr. Ir. Hendra Kusuma, M. Eng.Sc
Advisor 2nd : Dr. Ir. Totok Mujiono, M.Kom

ABSTRACT

The safe storage use to store important items, and generally still uses a conventional key where there is the possibility of missing or stolen keys. In this Final Project to make a security system based on sound recognition safe storage where voice is used as a password to open a valuable cabinet and enter the password on the keypad as a second security after the sound. This tool is able to store three different voice users, and equipped with database facilities to store the use of the cabinet activity is sent on google sheet. Voice processing using STM32F407VG using V3 voice recognition module. The result of this final project is able to recognize the sound of valuable cabinet users with different words from three users with the maximum acceptable sound intensity of 94.6 dB and the noise intensity level does not exceed 85.5 dB. The result of the speech recognition on the processed module is 80% of the twenty attempts. And can record the user time in the database in Real-Time.

Keywords **:** *STM32F407VG, Voice Recognition Module, Safe Storage.*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN SISTEM KEAMANAN LEMARI BENDA BERHARGA BERBASIS MODUL PENGENALAN SUARA

Dengan selesainya Tugas Akhir ini penulis menyampaikan terimakasih sebesar – besarnya kepada:

1. Kedua orang tua atas limpahan doa bagi penulis
2. Bapak Dr. Ir. Hendra Kusuma, M.Eng. Sc dan bapak Dr. Ir. Totok Mujiono, M.Kom selaku dosen pembimbing pertama dan kedua yang telah meluangkan waktu dan tenanganya untuk dapat membimbing penulis dalam menyelesaikan Tugas Akhir ini.
3. Seluruh staff dan pengajar departemen Teknik Elektro FTE – ITS.
4. Semua pihak yang telah banyak membantu untuk menyelesaikan Tugas Akhir ini.
5. Harapan kami sebagai penulis adalah terselesaikannya.

Tugas Akhir ini dapat bermanfaat bagi kami serta pembaca. Sadar atas keterbatasan yang dimiliki oleh penulis karena hasil dari Tugas Akhir ini belum sempurna. Demikian penulis berusaha semaksimal mungkin dan penulis mengharapkan pintu maaf serta saran dan kritik yang membangun.

Surabaya, 2 Juli 2018

Gusti Paring

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

-----Halaman ini sengaja dikosongkan----

DAFTAR GAMBAR

HALAMAN

DAFTAR TABEL

HALAMAN

-----Halaman ini sengaja dikosongkan-----

BAB I
PENDAHULUAN

1.1 Latar Belakang

Lemari benda berharga merupakan sebuah alat penyimpanan, pada saat ini lemari benda berharga masih banyak yang menggunakan kunci konvensional untuk dapat mengaksesnya. Penggunaan kunci konvensional ini kurang menguntungkan karena dapat hilangnya kunci dan adanya tindakan kriminal. Dewasa ini suara banyak digunakan pada peralatan elektronik.

Oleh karena itu dibuatnya tugas akhir ini dengan sistem keamanan lemari benda berharga berbasis modul pengenalan suara, dengan masukkan *password* menggunakan *keypad* sebagai keamanan kedua. Dalam pembuatan lemari benda berharga ini menggunakan STM32F407VG sebagai kontroler dan Modul Pengenalan Suara V3. Maka dengan adanya sistem keamanan suara ini pengguna akan merasa tenang ketika menyimpan barangnya dalam lemari benda berharga. Fasilitas pada lemari benda berharga ini diantaranya terdapat pengiriman pada basis data yang menunjukkan aktifitas dari lemari benda berharga. Sebelum lemari benda berharga digunakan untuk menyimpan maka pengguna harus memasukkan suara terlebih dahulu, lalu suara akan akan disimpan pada modul pengenalan suara V3 dan dapat dikenali. Ketika pengguna ingin mengambil barang maka akan terjadi pengenalan suara yang disimpan dengan yang dimasukkan oleh pengguna. Setelah adanya proses membukanya lemari sistem akan otomatis mencatat hari, tanggal, dan waktu pada google sheet. Hasil yang diperoleh adalah sebuah lemari benda berharga yang handal, kuat dan aman serta memudahkan pengguna dalam pengoperasiannya.

1.2 Permasalahan

Dari berbagai latar belakang tersebut, terdapat beberapa permasalahan, diantaranya adalah

1. Bagaimana mengenal dan mengidentifikasi perintah suara pengguna lemari.
2. Bagaimana cara memproses perintah suara dan mencocokan dengan data-data suara yang telah ada di basis data suara hasil pelatihan. Sehingga perintah tersebut dapat membuka lemari.

1.3 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut :

1. Jarak mikrofon dengan orang yang berbicara kurang dari 5 cm.
2. Masukkan suara juga berfungsi sebagai *password* yang hanya diketahui satu orang.
3. Penyimpanan suara tidak melebihi dari 3 orang.
4. Menggunnakan STM32F4 dan Modul Pengenalan Suara V3.
5. Pelatihan suara menggunakan Arduino.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut :

1. Dapat mengenali jenis warna suara dengan intonasi suara yang berbeda, dimana proses identifikasi suara tersebut akan dilaksanakan dengan modul pengenalan suara V3
2. Memproses dan mengenali data suara dengan modul pengenalan suara agar sesuai dengan yang tersedia pada basis data.

1.5 Metodologi Penelitian

Perancangan lemari benda berharga dengan sistem keamanan berbasis pengenalan suara, terbagi menjadi empat tahapan, yaitu studi literatur, perancangan sistem, uji coba dan hasil pengujian, serta penyusunan laporan.

Pada tahap studi literatur, dilakukan pencarian literatur buku maupun kumpulan makalah dan jurnal yang mengarah pada topik yang dibahas. Tahapan ini dilakukan untuk mengumpulkan informasi tentang pengolahan sinyal suara dengan metodenya.

Selanjutnya pada perancangan sistem, dibuat suatu program dengan *software* Keil V5 yang berfungsi untuk menjalankan proses melalui STM32F407VG. Pada tahapan ini membahas tentang *software* maupun *hardware.*

Setelah itu dilakukan pengujian alat. Pada tahapan ini menguji apakah sistem keseluruhan sudah berjalan sesuai dengan yang diinginkan.

1.6 Sistematika Laporan

Pembahasan tugas akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Pada bab pendahuluan, menjelaskan mengenai latar belakang pemilihan topik, perumusan masalah dan batasannya. Bab ini juga membahas mengenai tujuan penelitian, metodologi, sistematika laporan, dan relevansi dari penelitian yang dilakukan.

Bab II Tinjauan Pustaka Untuk Pengolahan Sinyal Suara

Penjelasan metode pengolahan sinyal dan aplikasi dari pengolahan sinyal untuk pengenalan suara pada lemari benda berharga.

Bab III Perancangan *Hardware* dan *Software* Pada Lemari Benda Berharga

Pembahasan yang dilakukan pada bab ini, mengenai perancangan sistem secara keseluruhan serta perancangan dari *Hardware.*

Bab IV Pengujian *Hardware* dan Simulasi Sistem

Hasil dari uji coba program yang dibuat pada Keil V5 dan uji coba *Hardware* dibahas secara lengkap pada bab ini.

Bab V Penutup

Pada bagian bab penutup, dibahas mengenai kesimpulan dan saran dari hasil pengujian.

1.7 Relevansi

Hasil yang diperoleh dari tugas akhir ini adalah terciptanya sebuah lemari penyimpan benda berharga yang mempunyai fitur keamanan berbasis pengenalan suara pengguna yang diharapkan menjadi suatu kontribusi yang berarti bagi pengembangan lemari benda berharga.

-----Halaman ini sengaja dikosongkan-----

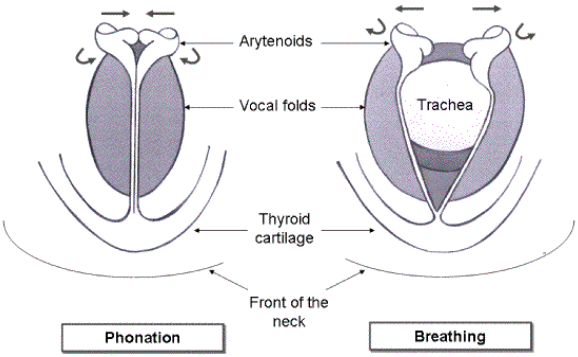
BAB II
TEORI DASAR

2.1 Karakteristik Suara [1]

Ketika seseorang berbicara maka akan mengeluarkan udara dari paru – paru. Udara lewat di antara lipatan vokal, yang merupakan jaringan otot di laring. Jika mendapat tekanan udara yang tepat, lipatan akan bergetar pada frekuensi akustik. Ini berarti terdapat katup yang berosilasi untuk membiarkan hembusan aliran udara ke saluran vokal pada beberapa frekuensi. Secara teknis berarti memindahkan kartilago arytenoid lebih dekat dari pada posisi bernafas, yang membawa lipatan vokal lebih dekat satu sama lain, yang disebut adduksi. Bukaan apeture yang berkurang diantara lipatan ini disebut glotis. Dibandingkan dengan posisi bernafas, glotis yang sempit membatasi aliran udara, pada keadaan itu akan terjadi penurunan tekanan yang melewati laring. Penurunan tekanan yang lebih tinggi berarti bahwa kecepatan udara melalui glotis akan tinggi, namun aliran volume (dalam liter per detik) akan berkurang, seperti pada Gambar 2.1.

Tekanan yang bekerja di bawah lipatan vokal akan cenderung memaksa ke atas dan terpisah. Perbedaan tekanan dapat mempercepat udara melalui glotis untuk menghasilkan udara berkecepatan tinggi. Aliran udara yang cepat melalui glotis akan menciptakan isapan. Hisapan ini akan menarik lipatan kembali, seperti halnya ketegangan pada lipatan suara. Efek bolak – balik ini cenderung membangkitkan skilus menutup dan membuka lipatan yang melekat dan akan dibantu oleh elastisitas lipatan vokal. Otot tidak secara langsung menggetarkan lipatan vokal yang merupakan efek pasif. Namun, otot akan berkontribusi pada kontrolnya, dengan menentukan banyak lipatan yang disatukan. Pada hal ini suara akan berada pada nada yang tetap , yang berati lipatannya bergetar secara teratur dan berkala.

Frekuensi berbicara biasanya 100 sampai dengan 400 Hz. Untuk menyanyi rentangnya bisa berkisar 60 Hz sampai 1500 Hz, tergantung pada jenis suara. Kecepatan suara sekitar 340 m/s, jadi panjang gelombang fundamental biasanya 1 sampai 3 meter, namun bisa sesingkat 0.3 m atau kurang untuk nyanyian bernada tinggi.

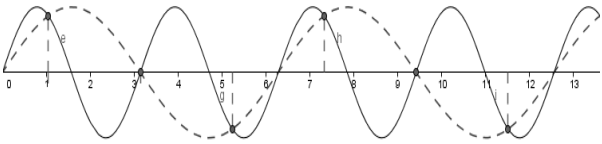


Gambar 2.1 Kondisi *Trachea* pada saat berbicara [1]

2.2 Sampling Sinyal [2]

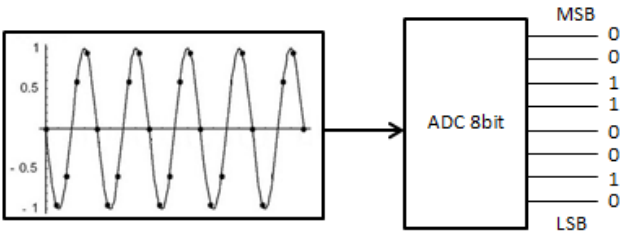
Ketika sampling gelombang anlog pada periode T (detik), frekuensi sampling (*sampling rate)* adalah 1/T Hz (Hertz-*cycle* per detik). Bentuk gelombang digital akan mewaliki dari bentuk gelombang analog, maka dari itu sampling rate harus lebih besar dari pada frekuensi tertinggi pada sinyal analog. Jika frekuensi sampling F (Hz), maka frekuensi yang terbesar adalah F/2 Hz (disebut dengan *Nyquist Frequency).*

Pada Gambar 2.2 menunjukkan bahwa gelombang sinus yang diambil pada interval tetap lebih besar dari pada satu setengah gelombang. Untuk representasi pada dua gelombang yang menjadi satu disebut dengan aliasing. Jika frekuensi yang lebih tinggi dari satu setengah frekuensi sampling yang didapat pada gelombang suara, maka terdapat kemungkinan akan terjadinya aliasing. Untuk menghindari aliasing, sinyal analog suara dapat terlebih dahulu difilter sebelum menjadikan sinyal tersebut ke digital. Filter yang dapat digunakan adalah antialiasing, band-pass, low-pass, presampling filter.



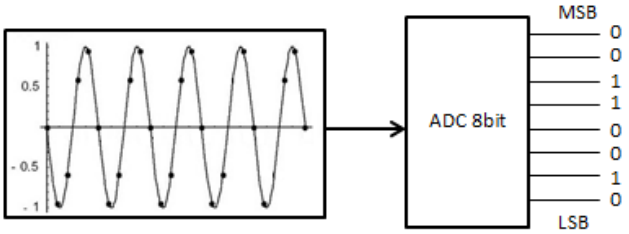
Gambar 2.2 Gelombang Sinus Dengan Aliasing

2.3 Kuantisasi ADC 8-bit



Gambar 2.3 Kuantisasi ADC 8-bit

Analog to Digital Converter (ADC) digunakan untuk mengubah sinyal analog menjadi sinyal digital. Dengan input sinyal suara yang dimana mempunyai sinyal digital akan menjadi digital yaitu dalam bentuk biner lalu akan disimpan pada memory yang ada pada kontroller. Adapun yang dimaksud kuantisasi 8-bit adalah jika suatu informasi harga sinyal analog diubah secara digital dan akan di representasikan dalam 8-bit data, seperti yang ditunjukkan pada



Gambar 2.3.

Nilai ADC 8-bit biasanya bernilai 0 sampai dengan 255 (00 – FF). Jika sinyal input memiliki amplitudo 1 volt dan nilai data 0.5V maka dapat diketahui output adc adalah 50 dengan rumus 500 mV / 10 mV. Lalu 50 dijadikan hexadecimal terlebih dahulu yaitu 32H. output pada ADC akan bernilai 00110010 biner.

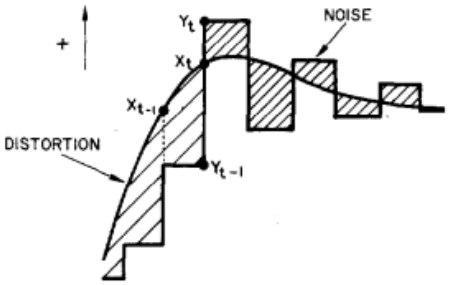
2.4 Modulasi Delta (Kuantisasi 1-bit) [3]

Modulasi delta adalah sistem komunikasi digital dengan kecepatan bit yang relatif rendah. Sistem ini digunakan untuk transmisi sinyal analog yang rentan terhadap gangguan sinyal (*noise)*. Modulasi delta ini bekerja

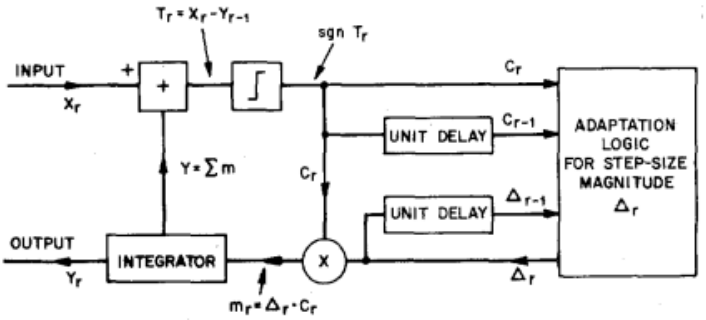
berdasarkan pada kuantisasi perubahan sinyal dari sampel ke sampel. Data yang dikirimkan akan dikurangi menjadi aliran data 1-bit.

Diagram blok modulasi delta dapat dilihat pada Gambar 2.4. Pada Gambar 2.4 terjadi pengurangan santara sinyal frekuensi rendah x(t) dan sinyal $x_s(t)$ yang akan menghasilkan perbedaan sinyal d(t), yang dimana $x_s(t)$ adalah sinyal refensi dari nilai sampling awal. d(t) dapat dinyatakan sebagai berikut.

$$d(t) = x(t) - x_s(t) \tag{2.1}$$



Gambar 2.4 Sinyal Modulasi Delta [3]



Gambar 2.5 Blok Diagram Modulasi Delta [3]

Perbedaan sinyal d(t) akan lebih besar dari pada nol, yang berarti bahwa sinyal referensi $x_s(t)$ lebih rendah dari pada x(t). Dengan demikian nilai sampling akan berkaitan dengan nilai sampling sebelumnya, yanga

akan dapat memudahkan perkiraan nilai sampling berikutnya. Blok diagram dari modulasi delta dapat dilihat pada Gambar 2.5.

2.5 Filter Digital [2]

Filter digital merupakan suatu sistem diskrit yang digunakan untuk memfilter (frekuensi) sinyal input digital menjadi sinyal output digital. Filter digital dikarakterisasi dengan persamaan beda koefisien konstan linier orde ke-N, selain itu dapat dinyatakan dalam respon impuls. Filter dikelompokkan menjadi dua yaitu *Finite Impulse Respon* (FIR) dan *Infinite Impulse Respon* (IIR). Filter FIR didesain dengan pendekatan filter digital ideal sedangkan filter IIR didesain dengan pendekatan filter analog.

Filter digital merupakan sistem linier *time-invariant* (LTI) yang melakukan proses dari input sinyal digital x(n) menjadi sinyal output digital y(n). Sistem LTI dapat dikarakterisasi dengan respon impuls h(n), fungsi sitem H(z) dan persamaan beda koefisien konstan.

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \tag{2.2}$$

2.5.1 Filter Finfinite Impulse Respon (FIR) [2]

Untuk mendesain filter FIR dilakukan pendekatan ke filter digital ideal. Untuk mendapatkan filter FIR adalah dengan membatasi panjang deretan respon impuls filter IIR. Filter FIR dapat direpresentaikan sebagai berikut.

$$h(n) = \begin{cases} h_d(n), & N_1 \leq n \leq N_2 \\ 0, & n\ lainnya \end{cases} \tag{2.3}$$

h(n) dapat dibentuk dengan mengalikan antara $h_d(n)$ dengan fungsi window w(n), sebagai berikut persamaan yang didapatkan.

$$h(n) = h_d(n) \,.\, w(n) \tag{2.4}$$

Jika menggunakan fungsi window *rectangular* maka respon impuls dari h(n) akan didapat sebagai berikut.

$$w(n) = \begin{cases} 1, & N_1 \leq n \leq N_2 \\ 0\ , & n\ lainnya \end{cases} \tag{2.5}$$

Jika dinyatakan sebagai transformasi fourier maka respon frekuensi $H(e^{j\omega})$ dari hasil desain filter merupakan konvolusi antara $H_d(e^{j\omega})$ dan $W(e^{j\omega})$ sebagai berikut.

$$H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta}).W\left(e^{j(\omega-\theta)}\right)d\theta = H_d(e^{j\omega}) * W(e^{j\omega}) \quad (2.6)$$

2.5.2 Filter Infinite Impulse Respon (IIR) [2]

Untuk mendesain filter IIR dibutuhkan banyak persyaratan yang harus terpenuhi. Respon impuls h[n] = 0, untuk n < 0. Pada keadaan stabil dapat dinyatakan sebagai berikut.

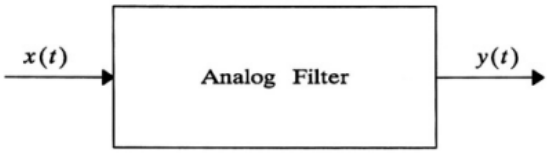
$$\sum_{-\infty}^{\infty}|h[n]| < \infty \quad (2.7)$$

Bila persamaan h(n) diubah pada transformasi-z (H(z)) maka terdapat syarat yang harus terpenuhi yaitu nilai $a_k \neq 0$, dan $M \leq N$. berikut persamaan H(Z) yang didapatkan.

$$H(Z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n} = \frac{\sum_{k=0}^{M} b_k z^{-1}}{\sum_{K=1}^{N} a_k z^{-1}} \quad (2.8)$$

2.6 Filter Analog [4]

Filter analog biasanya digunakan untuk menghilangkan noise pada sinyal. Biasanya filter analog tidak bergantung pada waktu (*time-invariant)* seperti yang ditunjukkan pada Gambar 2.6. Filter analog didefinisikan dalam domain frekuensi. Aplikasi dari filter analog adalah melwatkan satu sinyal dan menekan sinyal yang melebihi batas domain frekuensi. Terdapat banyak jenis filter analog yaitu *Low-pass, High-Pass, Band-Pass,* dan *Band-stop.*

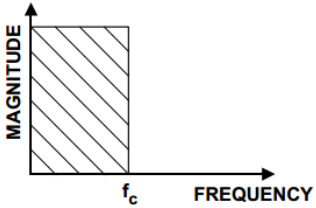


Gambar 2.6 Blok Diagram Filter Analog [4]

2.6.1 Low-Pass Filter (LPF) [4]

Low-pass filter adalah jenis filter yang melewatkan sinyal dengan frekuensi rendah dari frekuensi cut-off, dan akan menahan frekuensi yang lebih tinggi dari cut-off seperti pada Gambar 2.7. Untuk menghitung orde pada *Low Pass Filter* analog butterworth yaitu sebagai berikut.

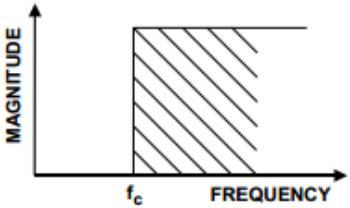
$$n = \frac{\log[(10^{-k1/10}-1)/(10^{-k2/10}-1)]}{2.\log\left(\frac{1}{\Omega}\right)} \quad (2.9)$$



Gambar 2.7 Respon *Low Pass Filter* [5]

2.6.2 High-Pass Filter (HPF) [4]

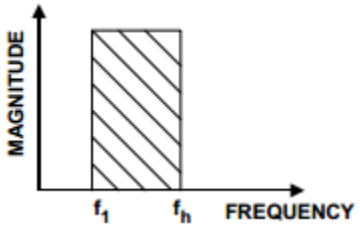
High-pass filter merupakan kebalikan dari Low-pass filter yaitu akan melewatkan sinyal dengan frekuensi tinggi dari cut-off dan akan menahan sinyal yang frekuensinya lebih rendah dari cut-off seperti pada Gambar 2.8.



Gambar 2.8 Respon High Pass Filter[4]

2.6.3 Band-Pass Filter (BPF) [4]

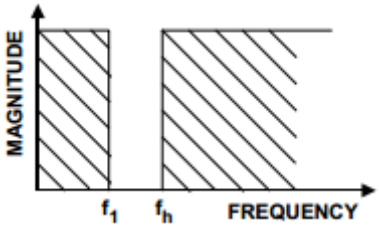
Band-pass filter merupakan gabungan dari low-pass dan high pass, yaitu akan melewatkan sinyal yang berada pada batas bawah dan batas atas, atau biasa disebut dengan frekuensi tengah seperti pada Gambar 2.9.



Gambar 2.9 Respon Band-Pass Filter[4]

2.6.4 Band-Stop Filter[4]

Band-Stop Filter merupakan kebalikan dari BPF yang dimana meloloskan batas atas dan juga batas bawah, dan frekuensi tengah akan ditahan seperti pada Gambar 2.10.



Gambar 2.10 Respon Band-Pass Filter[4]

2.7 Transformasi–Z [2]

Transformasi-Z merupakan suatau alat bantu pada analisis sinyal dan sistem waktu diskrit, terutama digunakan untuk analisa filter digital. Transformasi ini dapat digunakan untuk menyelesaikan persamaan beda koefisien konstan linier, mengevaluasi respon sistem *Linier Time-Invariant* (LTI) bila diberi sinyal masukan dan merencanakan filter digital linier. Transformasi-Z dari sinyal diskrit x(n) didefinisikan :

$$x(n) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \qquad (2.10)$$

Dimana $z = r\, e^{j\omega}$ yang mmerupakan variable untuk bilangan komplek. Nilai z agar X(z) merupakan konvergen jumlah didefinisikan sebagai daerah konvergensi bidang z. Transformasi-Z dapat ditinjau

sebagai transformasi Fourier diskrit (TFD) dari sinyal diskrit terbobot secara eksponensial.

$$\mathrm{x(z)} = \sum_{n=-\infty}^{\infty} x(n)z^{-n} = \sum_{n=-\infty}^{\infty} x(n)(re^{j\omega})^{-n} = \sum_{n=-\infty}^{\infty}(x(n)r^{-n})e^{-j\omega n} \quad (2.11)$$

2.8 Discrete Fourier Transform (DFT) [2]

Discrete Fourier Transform (DFT) adalah suatu metode yang digunakan untuk pemrosesan sinyal digital. DFT sendiri adalah suatu gambaran karakteristik spektrum periodik dari sample data. DFT memiliki spektrum garis yang mewakili periode sekuensial N. DFT dapat didefinisikan sebagai berikut.

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (2.12)$$

Untuk mendapatkan transformasi sinyal diskrit maka penjumlahhana pada persamaan harus bersifat konvergen, dan jika x(n) dijumlahkan secara absolut. Transformasi diskrit ini mempunyai periode 2π.

$$\sum_{n=-\infty}^{\infty}|x(n)| = S < \infty \quad (2.13)$$

2.9 Fast Fourier Transform (FFT) [2]

Fast Fourier Transform (FFT) adalah suatu metode yang mengambil sampel dalam waktu dan mengubahnya pada domain frekuensi. Proses dari FFT ini lebih cepat dan akurat dibandingkan dengan DFT. Pada FFT juga dapat menghemat untuk memori dan juga waktu pemrosesan. FFT dapat dituliskan sebagai berikut.

$$\mathrm{x(k)} = \sum_{n=0}^{N-1} x(n)e^{-j\frac{-2\pi}{N}} \quad (2.14)$$

Pada rumus diatas untuk $e^{-j\frac{-2\pi}{N}}$ dapat dituliskan dengan

$$cos\frac{2\pi}{N} - j\sin\frac{2\pi}{N} \quad (2.15)$$

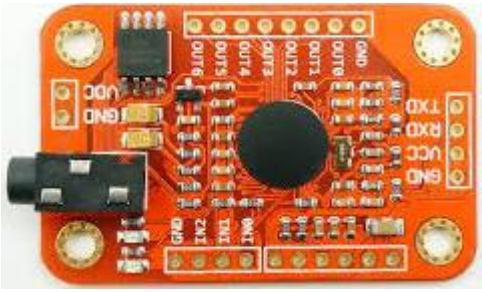
2.10 Mikrokontroller STM32 [5]

Mikrokontroller adalah suatu perangkat yang digunakan untuk mengatur jalannya sistem. Mikrokontroller sendiri terdiri dari *Central processing Unit* (CPU), memori, *General Port Input Output* (GPIO), *Analog to Digital Converter* (ADC), dan *Digital Signal Processing* (DSP) yang sudah terintegrasi pada mikrokontroller STM32. DSP digunakan untuk memproses sinyal mulai dari masuknya sinyal hingga mengubah sinyal dari domain waktu ke domain frekuensi. Operasi DSP pada STM32 dapat menggunakan dua format yaitu floating-point atau fixed-point. Mikrokontroller ini mempunyai *flash memory* sebesar 1-Mbyte dan 192-Kbyte RAM. *Clock* pada STM32F4 sebesar 168 MHz Gambar fisik dari STM32 seperti pada Gambar 2.11.

Gambar 2.11 Bentuk Fisik STM32 [5]

2.11 Modul Pengenalan Suara

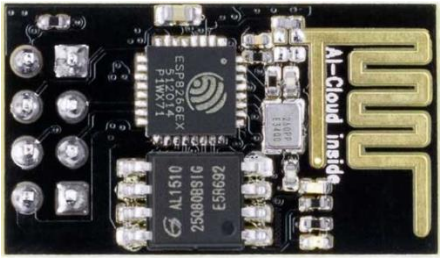
Modul pngenalan suara V3 merupakan suatu modul elektronika yang berfungsi untuk mengenali suara yang kemudian akan dimasukkan pada modul tersebut. Modul ini mempunyai dua komunikasi yaitu serial dan GPIO *built-in.* Modul ini dapat menerima 80 perintah suara dengan kecepatan durasi yaitu 1500ms, dan menyimpan 7 perintah suara yang dapat diakses. Modul ini membutuhkan *input* tegangan sebesar 5 volt. Data yang telah dikenali oleh modul akan dikirimkan melalui komunikasi serial dengan *baudrate* 9600 yang data tersebut berbentuk hexadecimal. Untuk tampilan fisik dari modul Pengenalan suara dapat dilihat pada Gambar 2.12.



Gambar 2.12 Bentuk Fisik Modul Pengenalan Suara

2.12 Modul Wifi

Modul Wifi berfungsi untuk menghubungkan antara device dengan internet dan membuat koneksi TCP/IP, serta berfungsi untuk menerima atau mengirim data melalui internet. Modul ini membutuhkan 3.3 volt sebagai sumber tegangannya. Modul ini memiliki MCU, Internal SRAM dan ROM, serta memiliki pin GPIO. Baudrate yang digunakan pada modul wifi dapat diubah sesuai dengan kebutuhan, baudrate pada pengaturan awal dibuat 115200. Berikut bentuk fisik dapat dilihat pada Gambar 2.13.



Gambar 2.13 Bentuk Fisik Modul WIFI

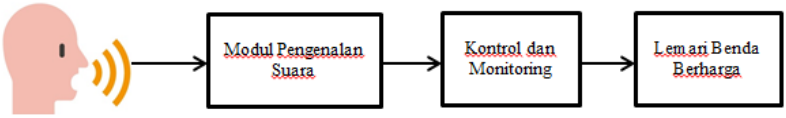
-----Halaman ini sengaja dikosongkan----

BAB III
PERANCANGAN SOFTWARE DAN HARDWARE

Pada bab ini akan dibahas perancangan suatu sistem baik dalam pembuatan *Software* maupun *Hardware* pada tugas akhir ini. Yang pertama menjelaskan tentang blok fungsional dari sistem lemari benda berharga, perancangan mekanik (*Hardware*) yang digunakan, perancangan elektrik yang berfungsi untuk menjalankan suatu sistem, dan pembuatan program pada Keil V5.

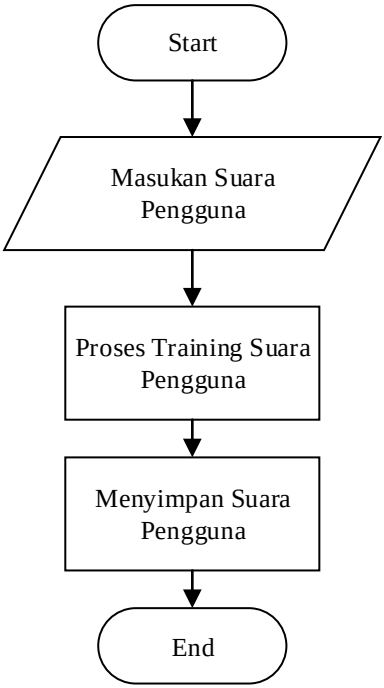
3.1 Blok Fungsional dan Diagram Alir Sistem

Pada tugas akhir ini menggunakan metode pengenalan suara untuk membuka lemari benda berharga untuk menghindari pemilik kehilangan kunci konvensional. Pada Gambar 3.1 dapat dilihat diagram blok fungsional yang menunjukkan jalannya dari sistem yang digunakan. Pertama suara pengguna akan masuk pada modul pengenalan suara V3 yang nantinya akan disimpan dan diproses oleh modul tersebut. Modul akan mengirimkan data HEX pada kontroller yang dimana menggunakan STM32F407VG, STM32 merupakan kontroller yang digunakan sebagai memproses data suara yang didapat dari modul dan membuka dan menutup dari lemari benda berharga, kontroler juga berfungsi sebagai pengiriman data yang nantinya terdapat aktivitas dari lemari benda berharga pada google sheet, dengan begitu pengguna dapat memonitoring aktivitas lemari benda berharga. Lalu setelah pemrosesan data suara, kontroller akan membuka lemari benda berharga dengan cara mengaktifkan solenoid yang merupakan bagian dari lemari benda berharga, sebelum itu terdapat masukkan password yang berfungsi untuk keamanan kedua ketika suara dapat dikenali..



Gambar 3.1 Blok Fungsional Sistem

3.1.1 Proses Penyimpanan Suara



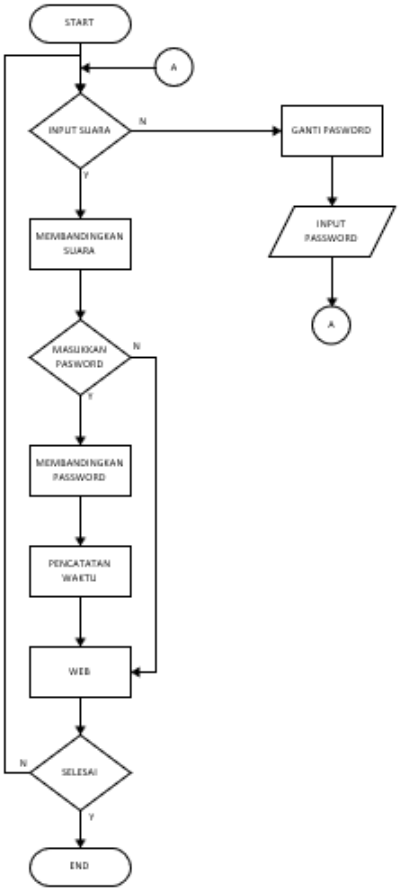
Gambar 3.2 Diagram Alir Penyimpanan Suara

Proses penyimpanan suara ini menggunakan modul *voice recognition* dan juga Arduino. Pada proses ini suara akan disimpan pada modul dan diubah pada bentuk HEX yang nantinya akan dikirimkan pada kontroler untuk dikenali. Berikut diagram alir untuk penyimpanan suara dapat dilihat pada Gambar 3.2.

3.1.2 Proses Pengenalan Suara

Proses pengenalan suara menggunakan STM32F4 yang dimana kontroler menerima data HEX dengan komunikasi serial. Data HEX yang didapat akan disimpan pada Array dan dibandingkan dengan suara yang lain. Setelah itu membuat *password* untuk dapat membuka lemari benda

berharga dan mencatat waktu yang akan dikirim pada web. Untuk lebih jelasnya dapat diihat pada diagram alir pada Gambar 3.3.



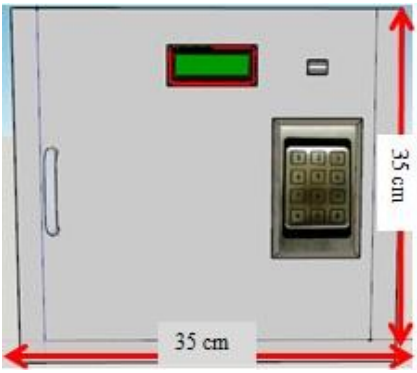
Gambar 3.3 Diagram Alir Pengenalan Suara

3.2 Perancangan Mekanik

Pada hal ini dilakukan perancangan mekanik dengan membuat sebuah lemari benda berharga untuk dapat berjalan sesuai sistem. Nantinya mekanik akan dihubungkan dengan elektrik. Lemari benda berharga ini berbentuk kubus sebesar 35cm x 35cm yang terbuat dengan

bahan *stainless steel*, dan terdapat 3 lubang yang berfungsi untuk LCD 20x4, *keypad* 4x4, dan mikrofon. Untuk pintu diberikan lubang LCD dengan ukuran 10cm x 4cm, sedangkan untuk mikrofon diberikan lubang melingkar dengan lebar diameternya 2cm dan diberikan lubang *keypad* 4x4 berbentuk persegi dengan ukuran 6cm x 6cm yang berfungsi untuk memasukkan *password*. Untuk rancangan awal seperti yang ada pada Gambar 3.4

Untuk merealisasikan desain awal dari Lemari Benda Berharga ditambahkan box elektrik pada bagian samping yang berfungsi sebagai tempat dari mikrokontroller dari lemari benda berharga. Box tersebut terbuat dari bahan akrilik dengan lebar 20x30cm, akan berisikan diantaranya mikrokontroller, driver relat, power supply, dan terminal. Berikut Gambar Realisasi dari Lemari benda berharga pada Gambar 3.5, Gambar 3.6.

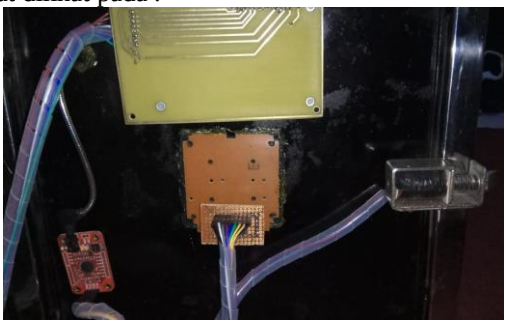


Gambar 3.4 Desain Awal Mekanik Tampak Depan

Gambar 3.5 Rancangan Mekanik Tampak Depan

Gambar 3.6 Rancangan Mekanik Tampak Samping

Pada balik pintu lemari benda berharga dipasang solenoid yang berfungsi untuk mengunci dan juga modul pengenalan suara serta rangkaian *pull down* untuk *keypad 4x4* dan rangkaian LCD 16x4, rangkaian tersebut dihubungkan menggunakan kabel melalui lubang kecil pada bagian samping yang terhubung dengan box elektrik. Untuk lebih jelasnya dapat dilihat pada .



Gambar 3.7 Rancangan Bagian Belakang Pintu Lemari

3.3 Perancangan Elektrik

Pada perancangan elektrik dilakukan pembuatan rangkaian yang berfungsi sebagai *Hardware* dari sistem, yaitu membuat shield dari STM32F407VG yang disambungkan dengan komponen yang

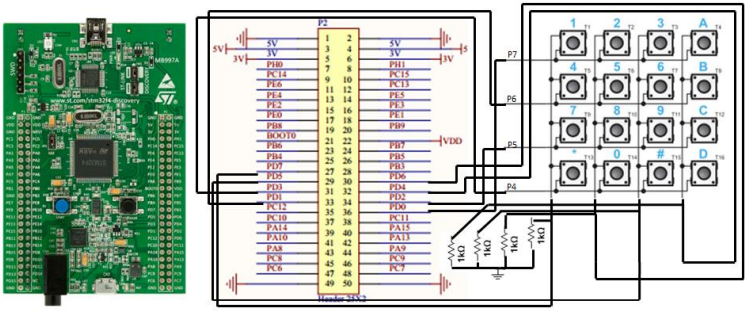
dibutuhkan. Pada hal ini STM32f407VG disambungkan dengan keypad 4x4, LCD 20x4, Modul Pengenalan Suara V3, relay, modul ESP8266 ESP-07.

3.3.1 Wiring Diagram Keypad

Untuk *keypad* membutuhkan 8 pin GPIO dan 1 pin GND. Yang dimana pin 1 sampai 2 disambungkan dengan GPIOD PD4 sampai dengan GPIOD PD7, sedangkan pin 5 sampai 8 disambungkan pada GPIOD PD0 sampai dengan GPIOD PD3. Sambungan dapat dilihat pada Tabel 3.1 dan Gambar 3.8.

Tabel 3.1 GPIO Keypad Dengan STM32F4

Keypad	STM32F407VG
PIN 0	GPIOD PD7 (output)
PIN 1	GPIOD PD6 (output)
PIN 2	GPIOD PD5 (output)
PIN 3	GPIOD PD4 (output)
PIN 4	GPIOD PD3 (input)
PIN 5	GPIOD PD2 (input)
PIN 6	GPIOD PD1 (input)
PIN 7	GPIOD PD0 (input)



Gambar 3.8 Wiring Diagram Keypad dan STM32F407

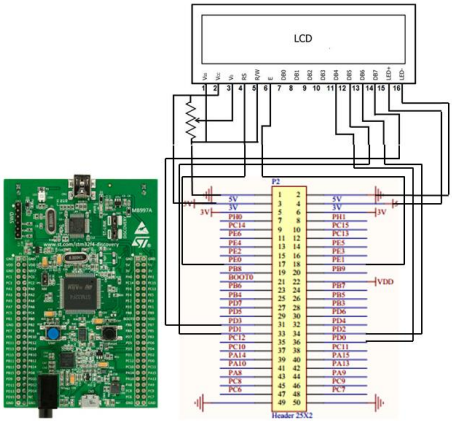
3.3.2 Wiring Diagram LCD 20x4

LCD 20x4 membutuhkan 10 pin yaitu VDD yang disambung pada sumber dengan diberikan potensio sebesar 10k untuk dapat mengatur kecerahan dari layar, VSS yang disambung pada GND, RS disambungkan dengan pin STM32F407 PE0 yang difungsikan sebagai output, E

disambungkan dengan pin STM32F407VG PE1 yang difungsikan sebagai output, D4, D5, D6, D7 disambungkan pada pin GPIO PB4 sampai dengan PB7 pada STM32. Untuk sambungan LCD dapat dilihat pada Tabel 3.2 dan Gambar 3.9.

Tabel 3.2 GPIO LCD 20x4 STM32F4

LCD 20x4	STM32F407VG
PIN VSS	PIN GND
PIN VCC	PIN 5 volt
PIN RS	GPIOE PE0 (output)
PIN E	GPIOE PE1 (output)
PIN D4	GPIOB PB4 (output)
PIN D5	GPIOB PB5 (output)
PIN D6	GPIOB PB6 (output)
PIN D7	GPIOB PB7 (output)
PIN B+	PIN 5 volt
PIN B-	PIN GND



Gambar 3.9 Koneksi LCD 20x4 dan STM32F407VG

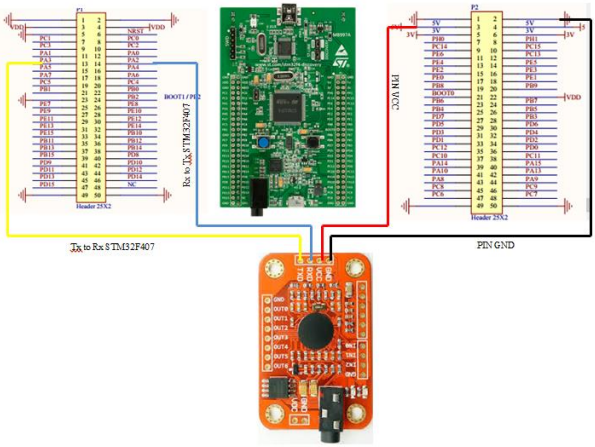
3.3.3 Wiring Diagram Modul V3

Modul Pengenalan Suara menggunakan komunikasi serial yang menghubungkan Tx dan Rx dari modul dengan STM32F407VG. Pin pada STM32 yang digunakan sebagai Tx dan Rx adalah pin PA2 dan PA3.

Berikut sambungan modul dengan STM32 dapat dilihat pada Tabel 3.3 dan Gambar 3.10.

Tabel 3.3 GPIO Modul Dengan STM32F4

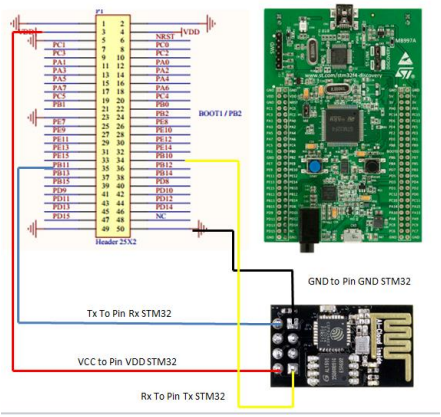
Modul V3	STM32F407VG
Tx	GPIOA PIN PA2 (Tx)
Rx	GPIOA PIN PA3 (Rx)
VCC	PIN 5 volt
GND	PIN GND



Gambar 3.10 Wiring Diagram Modul dan STM32F407VG

3.3.4 Wiring Diagram ESP8622 07

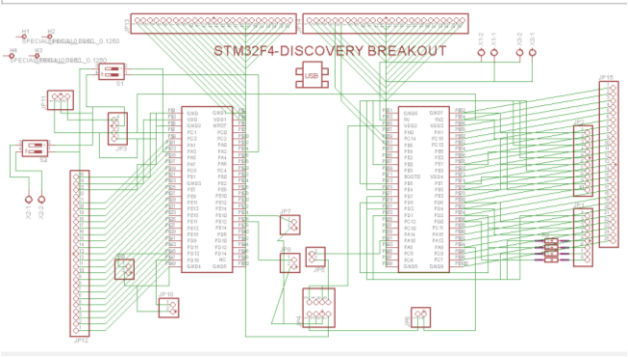
Modul wifi ESP8622 ini digunakan untuk mengirimkan data pada Google Sheet. Nantinya akan disambungkan pada USART yang ada pada STM32F407VG, pada Pin PB10 sebagai Rx dan PB11 sebagai Tx. Berikut Wiring Diagram dapat dilihat pada Gambar 3.11.



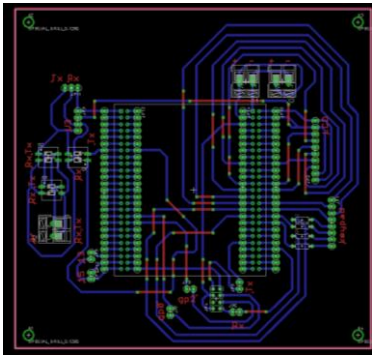
Gambar 3.11 Wiring Diagram ESP8266 dan STM32F407VG

3.3.5 Wiring Diagram Keseluruhan Sistem

Wiring keseluruhan sistem menghubungkan semua komponen menjadi satu. Pertama menghubungkan STM32F407VG dengan modul pengenalan suara V3, lalu Keypad, LCD, Driver Relay,dan modul ESP8266-07. Untuk mempermudah dalam menghubungkan modul tersebut pada wiring diagram keseluruhan ini membuat *Shield* dengan *Software* Eagle. Pembuatan *shield* untuk meminimalisir adanya kabel dan lebih praktis untuk wiringnya. Berikut wiring pada software Eagle dapat dilihat pada. Untuk Skematik dan Routing pada Eagle dapat dilihat pada Gambar 3.12 dan Gambar 3.13.



Gambar 3.12 Skematik Pada Eagle



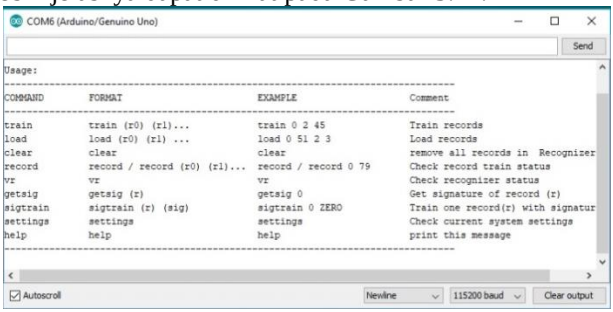
Gambar 3.13 Routing Pada Eagle

3.4 Fitur Pada Lemari Benda Berharga

Pada lemari benda berharga ini memiliki beberapa fitur yang diantaranya adalah pengenalan suara melalui mikrofon dengan maksimal tiga orang pengguna, pengamanan kedua dengan masukkan *password* yang berbeda setiap pengguna serta penggantian *password* maupun pada keadaan darurat, tampilan LCD yang bertujuan untuk melihat kondisi dari lemari benda berharga.

3.4.1 Fitur Pelatihan Suara

Fitur pelatihan suara ini berfungsi untuk melatih suara pada modul pengenalan suara yang dimana nantinya akan difungsikan sebagai masukkan password ketika akan menggunakan lemari benda berharga. Fitur ini menggunakan Arduino untuk dapat melatih suara pengguna. Untuk lebih jelasnya dapat dilihat pada Gambar 3.14.



Gambar 3.14 Pelatihan Suara

3.4.2 Fitur Pengenalan Suara

Fitur pengenalan suara ini berfungsi sebagai *password* bagi pengguna yang akan menyimpan barang pada lemari benda berharga. Suara pengguna akan dikenali oleh modul pengenalan suara berdasarkan intonas dan keras suara pada pelatihan suara. Untuk pengoprasian dapat dilihat pada tulisan yang ditampilkan LCD20x4, jika terdapat masukkan suara maka dapat memasukkan suara pengguna sesuai pelatihan suara yang disimpan pada modul pengenalan suara. Untuk lebih jelasnya dapat dilihat pada Gambar 3.15.

Gambar 3.15 Perintah Masukkan Suara

3.4.3 Fitur Keamanan Kedua

Fitur keamanan kedua berfungsi sebagai tambahan pengamanan apabila suara pengguna dapat ditiru oleh orang lain. Fitur tersebut menggunakan keypad sebagai masukkan password ketika pengguna akan membuka lemari benda berharga. Pengguna akan diminta memasukkan password ketika suara telah dikenali. Password awal dari pengguna dapat dirubah sesuai dengan keinginan pengguna. Untuk lebih jelasnya dapat dilihat pada Gambar 3.16.

Gambar 3.16 Perintah Masukkan Password

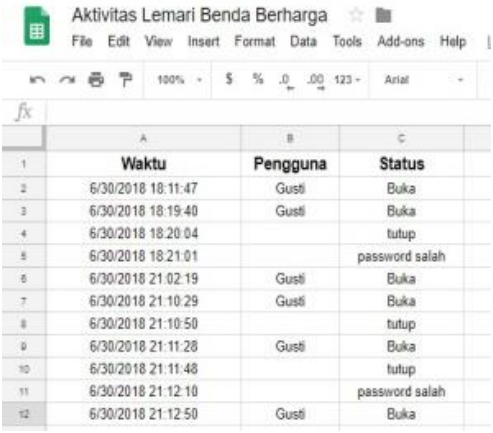
3.4.4 Fitur Keadaan Darurat

Pada fitur ini berguna untuk mereset password dan membuka lemari pada keadaan yang darurat. Apabila pengguna lupa cara pengucapan dari suara yang telah dilatih, ataupun lupa dengan password yang dimasukkan, maka pengguna dapat melakukan langkah tersebut untuk mereset password baru dan dapat membukanya dengan syarat memasukkan 10 angka yang diberikan pada pengguna. Untuk lebih jelasnya dapat dilihat pada Gambar 3.17.

Gambar 3.17 Perintah Memasukkan *Emergency Password*

3.4.5 Fitur Aktivitas Lemari Benda Berharga

Fitur ini berfungsi untuk dapat melihat pengguna yang mengakses lemari tersebut. Pengiriman data akan dikirimkan pada gmail orang yang berhak diberikan akses. Aktivitas pada basis data akan menunjukkan status lemari benda berharga pada kondisi tertutup, terbuka, ataupun telah diakses oleh orang lain. Untuk lebih jelasnya dapat dilihat pada .



	A	B	C
1	**Waktu**	**Pengguna**	**Status**
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:20:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup
9	6/30/2018 21:11:28	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:10		password salah
12	6/30/2018 21:12:50	Gusti	Buka

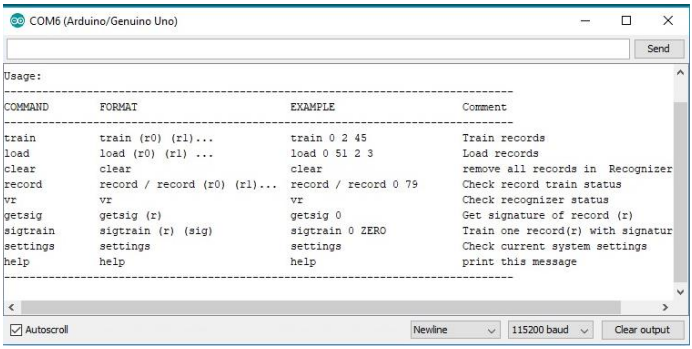
Gambar 3.18 Tampilan Basis Data pada Google Sheet

3.5 Perancangan Software

Pada perancangan software ini meliputi pembuatan program untuk penyimpanan data suara menggunakan Arduino, pengambilan data dan pengenalan suara menggunakan STM32F407VG.

3.5.1 Program Penyimpanan Suara

Pada penyimpanan suara menggunakan arduino dengan modul pengenalan suaraV3 yang mana menghubungkan Tx, Rx, VCC, dan GND. Berikut penyimpanan suara menggunakan arduino. Untuk dapat memasukkan pelatihan suara dapat mengetik *sigtrain* sedangkan untuk memeriksa bahwa sudah berapa perintah yang disimpan kita dapat mengetik *load* pada *Serial Monitor*. Format untuk menuliskan sigtrain yaitu sigtrain 0(0 = sebagai index disimpannya perintah suara) on(on = sebagai tanda bahwa pada index 0 merupakan perintah on)Berikut tampilan untuk menyimpan suara pada serial monitor arduino pada Gambar 3.19 . dan program dapat dilihat pada Lampiran A.



Gambar 3.19 Serial Monitor Pelatihan Suara Pada Arduino

3.5.2 Pembuatan Program Untuk Pengambilan Data Suara

Pengambilan data suara menggunakan STM32F407VG dengan mengirimkan perintah LOAD dalam bentuk HEX ke modul V3. Untuk pengiriman pertama perintah *clear recognition* yaitu berfungsi untuk menghentikan pengenalan suara yang sebelumnya agar tidak tertumpuk. Setelah itu menunggu balasan yang menandakan bahwa *clear recognition* berhasil, lalu kita bisa mengirimkan perintah LOAD dalam bentuk HEX yang mana dibedakan dengan letak *index* dari penyimpanan suara.

Berikut program pengambilan data suara. Untuk program lebih jelasnya dapat dilihat pada Lampiran.

```
pData[0]=0xAA;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x02;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x31;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(1000);

pData[0]=0xAA;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x03;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x30;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x00;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(1000);

pData[0]=0xAA;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x03;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x30;
```

```
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x01;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(1000);
pData[0]=0xAA;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x03;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x30;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x02;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
```

3.5.3 Pembuatan Program untuk Pengenalan Suara

Program pengenalan data suara ini menggunakan LED untuk tanda bahwa suara telah dikenali. Dengan mengambil data suara dapat dilakukan untuk membedakan suara index 0, 1 dan 2. Sama halnya pada program pengambilan data, device mengirim HEX ke modul lalu akan didapat data HEX yang nantinya akan dibandingkan dengan data yang lain.Berikut program untuk pengenalan suara.

```
while (1)
  {
zz=0;
yy=0;
HAL_UART_Receive(&huart2, rBuf, sizeof(rBuf), 1000);
for (int i = 0;i<12;i++)
if(rBuf[i]==on[i]){
zz=zz+1;
}
```

```
else if(rBuf[i]==off[i]){
      yy=yy+1;
      }
      if (zz>8){
      HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_SET);
      }
      if (yy>4){
      HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_RESET
);
      }
```

3.5.4 Pembuatan Program Untuk Keypad Dan LCD 20x4

Program keypad dan LCD ini berfungsi untuk dan pasword serta interface ketika proses dapat dilanjutkan. Petama dilakukan inisialisasi keypad dan juga LCD. Lalu menampilkan angka keypad yang ditekan pada LCD.

```
      while (1)
        {
          LCD1602_print("Pressed Key : ");
          LCD1602_setCursor(2,1);
          Keypad4x4_ReadKeypad(mySwitches);
          for(uint8_t i=0; i<16; i++)
          {
          if(mySwitches[i])
          {
          LCD1602_print(Keypad4x4_GetChar(i));
          }
          }
          HAL_Delay(100);
          LCD1602_clear();
        }
```

3.5.5 Pembuatan Program Google Script

Program pada Google Script bertujuan untuk dapat mengakses google sheet dengan alat, alat akan mengirimkan data yang diinginkan yang nantinya akan disimpan pada google sheet. Google script akan memberikan alamat API yang akan diakses oleh ESP8266. Berikut program pada google script.

```
function doGet(e){
  return handleResponse(e);
}
//Recieve parameter and pass it to function to handle
function doPost(e){
  return handleResponse(e);
}
function doGet(e){
  return handleResponse(e);
}
// here handle with parameter
function handleResponse(request) {
  var output  = ContentService.createTextOutput();
  //create varibles to recieve respective parameters
  var time = new Date();
  var user = '';
  var id = request.parameter.id;
  var status = '';
  var key = request.parameter.key;
  //open your Spread sheet by passing id
  var ss= SpreadsheetApp.openById(id);
  var sheet=ss.getSheetByName("Activity");
  /////
  if (key === 'A'){
    status = 'Buka';
    user = 'Gusti';
  }
  if (key === 'B'){
    status = 'Buka';
    user = 'Prayoga' ;
  }
  if (key === 'C'){
    status = 'Buka';
    user = 'Hendra Kusuma' ;
  }
  if (key === 'D'){
    status = 'tutup';
  }
  if (key === 'E'){
```

```
  status = 'password salah';
 }
 if (key ==='F'){
  status = 'Buka';
  user = 'Andre';
 }
 //add new row with recieved parameter from client
 var rowData = sheet.appendRow([time,user,status]);
 var callback = request.parameters.callback;
 if (callback === undefined) {
  output.setContent(JSON.stringify({
   status: 'Success'}));
 } else {
  output.setContent(callback + "(" + JSON.stringify("Success") + ")");
 }
 output.setMimeType(ContentService.MimeType.JSON);
 return output;
}
```

3.5.6 Pembuatan Program ESP8266

Modul ESP8266 berfungsi untuk menghubungkan device dengan internet. Data membuka dan menutup akan dicatat pada google sheet, dengan keterangan hari, tanggal, bulan, tahun, dan waktu. Untuk web menggunakan google sheet sebagai penyimpanan datanya. Berikut program untuk ESP8266 dengan menggunakan AT-Command.

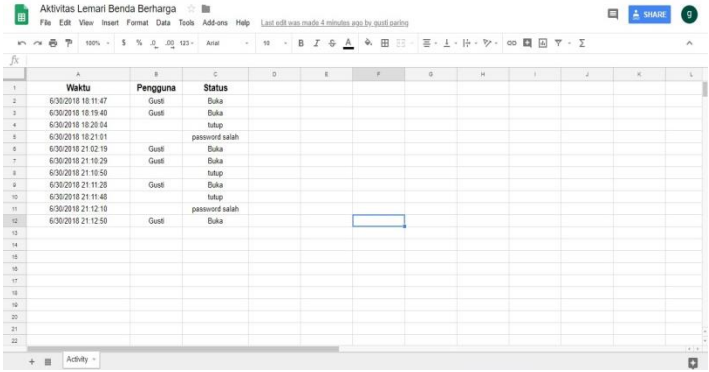
```
void sendA()
{
printf("AT+CIPMODE=0\r\n");
HAL_Delay(1000);
printf("AT+CIPMUX=1\r\n");
HAL_Delay(1000);
printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\r\n");
HAL_Delay(4000);
printf("AT+CIPSEND=4,131\r\n");
HAL_Delay(2000);
//print A
printf("GET
/pushingbox?devid=vFF6E7B22AE7CAAC&id=1IkBHk5A4bYIZFrvjZ
```

```
IxwrcZWBcX3U2zSNBwR3kA-e_0&key=A HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
}
void sendB()
{
printf("AT+CIPMODE=0\r\n");
HAL_Delay(1000);
printf("AT+CIPMUX=1\r\n");
HAL_Delay(1000);
printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\r\n");
HAL_Delay(4000);
printf("AT+CIPSEND=4,131\r\n");
HAL_Delay(2000);
//print B
printf("GET
/pushingbox?devid=v228B5E40CF7CBE1&id=1IkBHk5A4bYIZFrvjZI
xwrcZWBcX3U2zSNBwR3kA-e_0&key=A HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
}
void sendC()
{
printf("AT+CIPMODE=0\r\n");
HAL_Delay(1000);
printf("AT+CIPMUX=1\r\n");
HAL_Delay(1000);
printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\r\n");
HAL_Delay(4000);
printf("AT+CIPSEND=4,131\r\n");
HAL_Delay(2000);
//print C
printf("GET
/pushingbox?devid=vE1DB8545176D2A0&id=1IkBHk5A4bYIZFrvjZI
xwrcZWBcX3U2zSNBwR3kA-e_0&key=A HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
}
void sendD()
{
printf("AT+CIPMODE=0\r\n");
HAL_Delay(1000);
```

```
printf("AT+CIPMUX=1\r\n");
HAL_Delay(1000);
printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\r\n");
HAL_Delay(4000);
printf("AT+CIPSEND=4,131\r\n");
HAL_Delay(2000);
//print D
printf("GET
/pushingbox?devid=v4C357AA26277509&id=1IkBHk5A4bYIZFrvjZIx
wrcZWBcX3U2zSNBwR3kA-e_0&key=A HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
}
void sendE()
{
printf("AT+CIPMODE=0\r\n");
HAL_Delay(1000);
printf("AT+CIPMUX=1\r\n");
HAL_Delay(1000);
printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\r\n");
HAL_Delay(4000);
printf("AT+CIPSEND=4,131\r\n");
HAL_Delay(2000);
//print E
printf("GET
/pushingbox?devid=vD4FD9E7AC89160D&id=1IkBHk5A4bYIZFrvjZ
IxwrcZWBcX3U2zSNBwR3kA-e_0&key=A HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
}
void sendF()
{
printf("AT+CIPMODE=0\r\n");
HAL_Delay(1000);
printf("AT+CIPMUX=1\r\n");
HAL_Delay(1000);
printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\r\n");
HAL_Delay(4000);
printf("AT+CIPSEND=4,131\r\n");
HAL_Delay(2000);
//print E
```

```
printf("GET
/pushingbox?devid=vD4FD9E7AC89160D&id=1IkBHk5A4bYIZFrvjZ
IxwrcZWBcX3U2zSNBwR3kA-e_0&key=A HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
}
```

Berikut tampilan pada google sheet yang akan digunakan sebagai *database* pada Gambar 3.20 .



	A	B	C
1	**Waktu**	**Pengguna**	**Status**
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:26:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup
9	6/30/2018 21:11:28	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:10		password salah
12	6/30/2018 21:12:50	Gusti	Buka

Gambar 3.20 Tampilan Basis Data Pada Google Sheet

3.5.7 Pembuatan Program Keseluruhan

Pada program keseluruhan merupakan gabungan dari program keypad, LCD, modul V3, dan ESP. Pada program ini akan berjalan sesuai dengan diagram alir pada . Untuk program dapat dilihat pada Lampiran A.

-----Halaman ini sengaja dikosongkan----

BAB IV
PENGUJIAN LEMARI BENDA BERHARGA

Bab ini dibagi menjadi empat bagian. Pada bagian pertama akan menunjukkan hasil dari pelatihan suara pada Arduino. Bagian kedua akan menjelaskan tentang pengambilan data dari modul dengan menggunakan STM32F407VG. Bagian yang ketiga menguji modul wifi dengan mengirim data ke web, dan bagian terakhir merupakan pengujian lemari benda berharga secara keseluruhan.

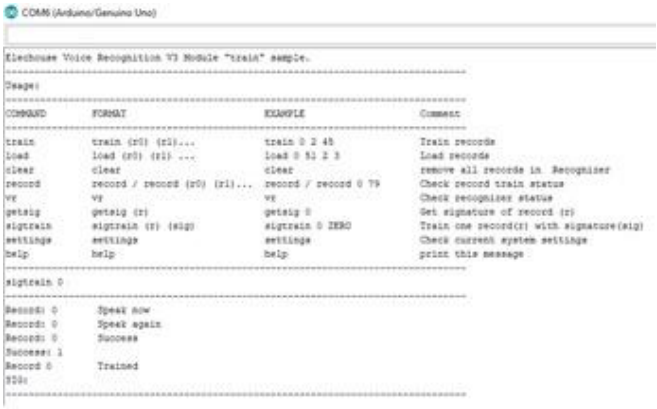
4.1 Pengujian Pelatihan Suara

Pada pengujian pelatihan suara menggunakan arduino sebagai kontroller untuk memudahkan pengolahan data tanpa harus menggunakan USB TTL. Suara pada index pertama menggunakan suara saya sendiri, suara pada index kedua dan ketiga menggunakan suara teman saya dengan mengatakan sesuatu yang berfungsi sebagai pasword pengguna. Berikut data pengguna serta kata yang disimpan pada modul pengenalan suara V3 dapat dilihat pada Tabel 4.1

Tabel 4.1 Data Pengguna

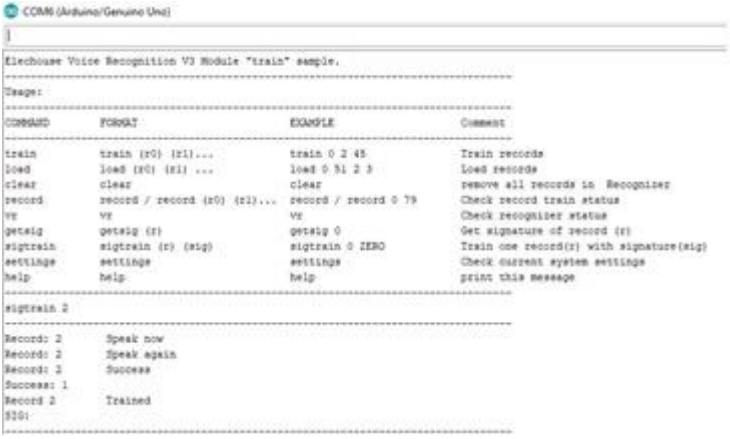
Index	Pengguna	Kata Password (kata yang disimpan)
0	Gusti	Paring
1	Prayoga	Yoga
2	Andre	Andre

Perintah yang digunakan pada serial monitor arduino adalah sigtrain (index) (*comment*). Berikut pengambilan data mulai dari *index* 1, 2 dan 3 pada arduino dapat dilihat pada Gambar 4.1, Gambar 4.2, Gambar 4.3



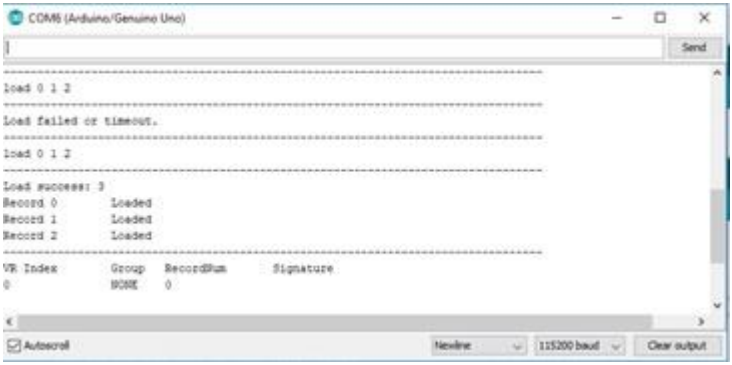
Gambar 4.1 Proses Sigtrain Index 0

Gambar 4.2 Proses Sigtrain Index 1



Gambar 4.3 Proses Sigtrain Index 2

Setelah pelatihan suara selesai, selanjutnya adalah memeriksa apakah suara yang telah disimpan benar sesuai dengan masukkan yang diberikan. Dengan mengetikkan load (nomor index). Berikut proses pemeriksaan pelatihan suara dapat dilihat pada serial monitor dapat dilihat pada Gambar 4.4.

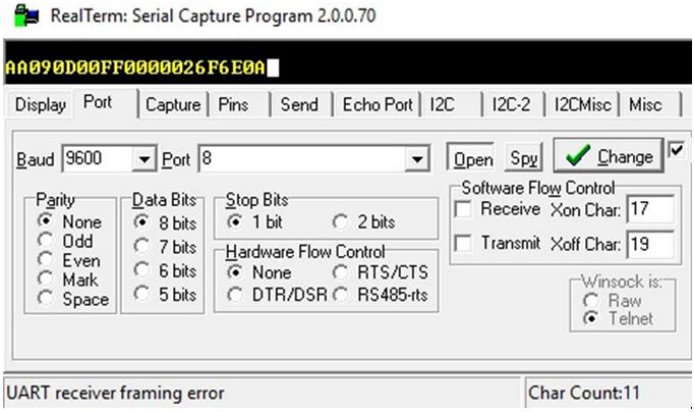


Gambar 4.4 Memuat Pelatihan Suara Pada Memori Modul V3

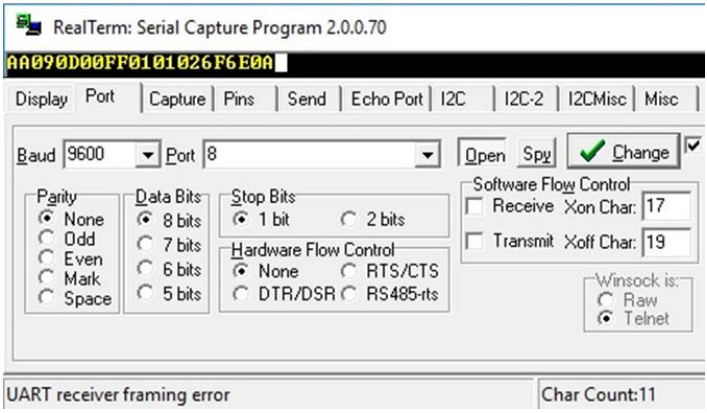
4.2 Pengambilan Data Suara

Proses pengambilan data suara digunakan untuk mengetahui nilai HEX yang diterima oleh kontroler, sehingga pada kontroller dapat membedakan kata pada index 0, 1, dan 2. Data suara yang didapat

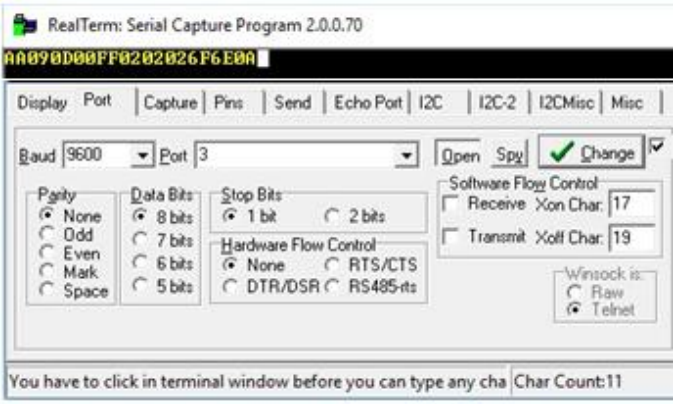
sepanjang 10 byte. Setiap suara yang disimpan memiliki data suara yang berbeda. Untuk pengambilan data menggunakan USB TTL yang dimana berfungsi untuk melihat data yang dikirim melalui Tx dari Modul, sedangkan untuk Tx dari STM32F407 dihubungkan dengan Rx dari modul pengenalan suara, dan pengambilan data menggunakan *software* serial yaitu Realterm. Berikut data suara setiap kata yang disimpan pada memori modul pengenalan suara V3 dapat dilihat pada Gambar 4.5, Gambar 4.6, Gambar 4.7.



Gambar 4.5 Data HEX Index 0



Gambar 4.6 Data HEX Index 1



Gambar 4.7 Data HEX Index 2

Dari pengambilan data menggunakan *software* Realterm dihasilkan seperti yang ada pada Tabel 4.2.

Tabel 4.2 Data HEX Setiap Suara

Index	Pengguna	Kata Password	Data HEX
0	Gusti	Paring	AA 09 0D 00 FF 00 00 02 6F 6E 0A
1	Prayoga	Yoga	AA 09 0D 00 FF 01 01 02 6F 6E 0A
2	Andre	Andre	AA 09 0D 00 FF 02 02 02 6F 6E 0A

4.2.1 Pengujian Pengenalan Suara

Dalam pengujian pengenalan suara ini dilakukan dengan tiga orang pengguna yang password katanya telah disimpan pada basis data. Tiga orang akan melakukan pengenalan kata sesuai dengan pasword setiap pengguna. Pengujian pengenalan suara menggunakan STM32F407VG yang dihubungkan dengan modul pengenalan suara V3. Cara pengujian yaitu setiap pengguna mencoba untuk menyebutkan kata yang telah disimpan sebanyak 20 kali. Pengujian dapat dilihat pada Tabel 4.3, Tabel 4.4, Tabel 4.5.

Tabel 4.3 Pengujian Pengguna Pertama

No	Pengguna	Kata yang diucapkan	Keberhasilan
1	Gusti	Paring	Berhasil
2			berhasil
3			berhasil
4			Berhasil
5			tidak
6			berhasil
7			berhasil
8			tidak
9			tidak
10			berhasil
11			berhasil
12			tidak
13			berhasil
14			berhasil
15			berhasil
16			berhasil
17			berhasil
18			berhasil
19			berhasil
20			berhasil
Presentase Keberhasilan(%)			80%

Tabel 4.4 Pengujian Pengguna Kedua

No	Pengguna	Kata yang diucapkan	Keberhasilan
1	Prayoga	Yoga	tidak
2			berhasil
3			tidak

No	Pengguna	Kata yang diucapkan	Keberhasilan
4	Prayoga	Yoga	berhasil
5			berhasil
6			berhasil
7			tidak
8			berhasil
9			tidak
10			tidak
11			berhasil
12			berhasil
13			berhasil
14			tidak
15			berhasil
16			berhasil
17			berhasil
18			berhasil
19			berhasil
20			berhasil
Presentase Keberhasilan(%)			70%

Tabel 4.5 Pengujian Pengguna Ketiga

No	Pengguna	Kata yang diucapkan	Keberhasilan
1	Andre	Andre	berhasil
2			berhasil
3			berhasil
4			berhasil
5			berhasil
6			berhasil

No	Pengguna	Kata yang diucapkan	Keberhasilan
7	Andre	Andre	berhasil
8			berhasil
9			berhasil
10			berhasil
11			berhasil
12			berhasil
13			berhasil
14			tidak
15			berhasil
16			berhasil
17			berhasil
18			berhasil
19			berhasil
20			tidak
Presentase Keberhasilan(%)			90%

4.3 Pengambilan Data Intensitas Suara

Pada pengambilan data ini dilakukan dengan menggunakan alat ukur intensitas suara yang ditunjukkan pada , dan dilakukan pada kamar dengan kondisi hening yang dimana intensitas suaranya38,6dB. Untuk meningkatkan tingkat kebisingan pada Gambar 4.8 pengambilan data ini menggunakan suara dari musik. Pengambilan intensitas suara ini untuk mengetahui suara yang dapat dikenali oleh modul pengenalan suara. Data dapat dilihat pada Tabel 4.6.

Gambar 4.8 Alat Ukur Intensitas Suara

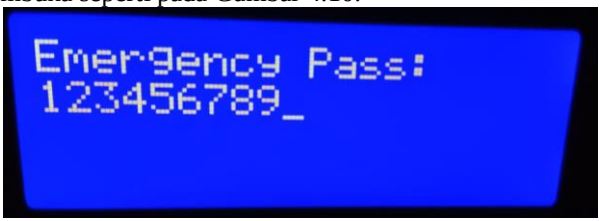
Tabel 4.6 Data Intensitas Suara

No	Intensitas Kebisingan	Kata yang diucapkan	Jarak	Intensitas Suara	Keberhasilan
1	58,9 dB	Paring	< 5 cm	67,4 dB	berhasil
2	63,1 dB			77,9 dB	berhasil
3	67,9 dB			80,2 dB	berhasil
4	72,7 dB			83,6 dB	berhasil
5	78,3 dB			85,3 dB	berhasil
6	81,2 dB			87,4 dB	berhasil
7	85,5 dB			88,9 dB	berhasil
8	88,4 dB			91,8 dB	tidak
9	91,7 dB			93,8 dB	tidak
10	93,2 dB			94,6 dB	tidak

4.4 Pengujian Keadaan Darurat

Pengujian ini berfungsi untuk merealisasikan keadaan darurat yang dimana ketika pengguna lupa dengan *password* suara ataupun lupa dengan *password* masukkan *keypad* maka dapat menggunakan fitur tersebut. Dengan memasukkan 10 angka yang telah ditentukan yaitu

1234567890 sebagai masukkan *password emergency*. Setelah pengguna memasukkan *password emergency* maka secara otomatis password yang telah disimpan akan mereset menjadi password awal dan lemari dapat membuka. Dengan menekan tombol D pada keypad maka akan muncul pilihan emergency yang digunakan untuk membuka lemari benda berharga dan mengatur ulang *password* pada pengaturan awal. Perintah masukkan *password emergency* ditunjukkan pada Gambar 4.9 dan loker akan membuka seperti pada Gambar 4.10.



Gambar 4.9 Perintah Memasukkan *Password*

Gambar 4.10 Lemari Membuka

4.5 Pengujian Pengiriman Data Ke Web

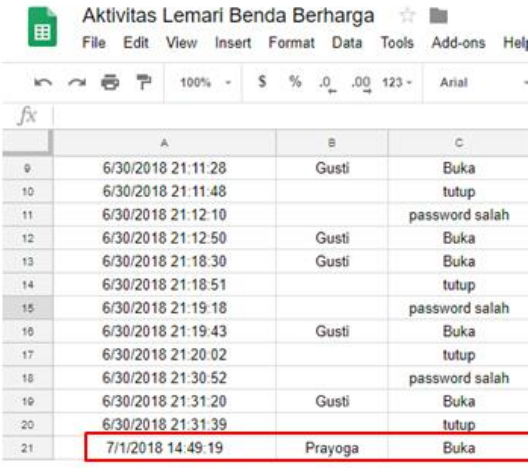
Pada pengujian ini untuk mengetahui pengiriman data ke web sesuai dengan yang diinginkan. Modul ESP8266 akan mengirim ke web dengan pencatatan waktu yang dimana pengguna dapat melihat aktifitas penggunaan dari lemari benda berharga. Ada 5 fungsi program untuk

mengirim yang pertama sendA() yang berfungsi untuk mengirimkan data Gusti dalam status membuka, dapat dilihat pada Gambar 4.11.

	Waktu	Pengguna	Status
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:20:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup
9	6/30/2018 21:11:28	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:10		password salah
12	6/30/2018 21:12:50	Gusti	Buka
13	6/30/2018 21:18:30	Gusti	Buka
14	6/30/2018 21:18:51		tutup
15	6/30/2018 21:19:18		password salah
16	6/30/2018 21:19:43	Gusti	Buka
17	6/30/2018 21:20:02		tutup

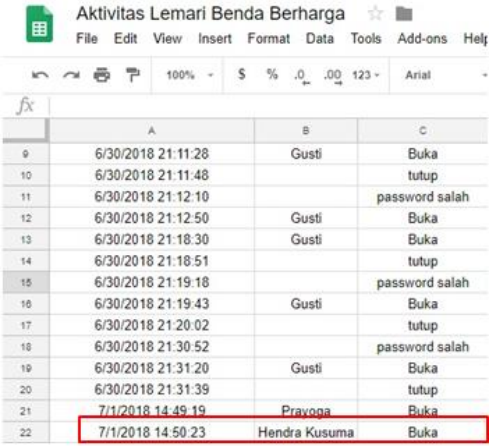
Gambar 4.11 Pengiriman Data Gusti

Kemudian Pengiriman sesuai program dengan fungsi sendB(), mengirim data Yoga dengan status Buka, dapat dilihat pada Gambar 4.12.



Gambar 4.12 Pengiriman Data Yoga

Kemudian Pengiriman sesuai program dengan fungsi sendC(), mengirim data Hendra Kusuma dengan status Buka, dapat dilihat pada Gambar 4.13



	A	B	C
9	6/30/2018 21:11:28	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:10		password salah
12	6/30/2018 21:12:50	Gusti	Buka
13	6/30/2018 21:18:30	Gusti	Buka
14	6/30/2018 21:18:51		tutup
15	6/30/2018 21:19:18		password salah
16	6/30/2018 21:19:43	Gusti	Buka
17	6/30/2018 21:20:02		tutup
18	6/30/2018 21:30:52		password salah
19	6/30/2018 21:31:20	Gusti	Buka
20	6/30/2018 21:31:39		tutup
21	7/1/2018 14:49:19	Prayoga	Buka
22	7/1/2018 14:50:23	Hendra Kusuma	Buka

Gambar 4.13 Pengiriman Data Hendra Kusuma

Kemudian Pengiriman sesuai program dengan fungsi sendD(), mengirim dengan status Tutup yang menandakan lemari benda berharga dalam kondisi tertutup, dapat dilihat pada Gambar 4.14

1	Waktu	Pengguna	Status
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:20:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup
9	6/30/2018 21:11:28	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:10		password salah
12	6/30/2018 21:12:50	Gusti	Buka
13	6/30/2018 21:18:30	Gusti	Buka
14	6/30/2018 21:18:51		tutup
15	6/30/2018 21:19:18		password salah
16	6/30/2018 21:19:43	Gusti	Buka
17	6/30/2018 21:20:02		tutup
18	6/30/2018 21:30:52		password salah
19	6/30/2018 21:31:20	Gusti	Buka
20	6/30/2018 21:31:39		tutup
21			

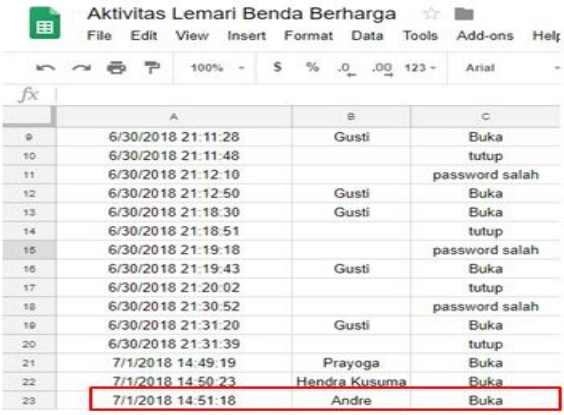
Gambar 4.14 Pengiriman Status Tutup

Kemudian Pengiriman sesuai program dengan fungsi sendE(), mengirim dengan status *Password salah* yang menandakan pengguna salah memasukkan *password*, dapat dilihat pada Gambar 4.15.

	Waktu	Pengguna	Status
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:20:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup
9	6/30/2018 21:11:28	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:10		password salah
12	6/30/2018 21:12:50	Gusti	Buka
13	6/30/2018 21:18:30	Gusti	Buka
14	6/30/2018 21:18:51		tutup
15	6/30/2018 21:19:18		password salah
16	6/30/2018 21:19:43	Gusti	Buka
17	6/30/2018 21:20:02		tutup
18	6/30/2018 21:30:52		password salah
19	6/30/2018 21:31:20	Gusti	Buka
20	6/30/2018 21:31:39		tutup
21			

Gambar 4.15 Pengiriman Status *Password* Salah

Kemudian Pengiriman sesuai program dengan fungsi sendF(), mengirim data Andre dengan status Buka, dapat dilihat pada Gambar 4.16.



Gambar 4.16 Pengiriman Data Andre

4.6 Pengujian Keseluruhan Sistem

Pengujian keseluruhan berfungsi untuk mengetahui apakah sistem dapat berjalan dengan baik dan handal. Dimulai dengan memasukkan suara sampai pengiriman status pada google sheet. Pertama memasukkan suara pengguna yang ingin diakses yang ditunjukkan pada Gambar 4.17.

Gambar 4.17 Masukkan Suara

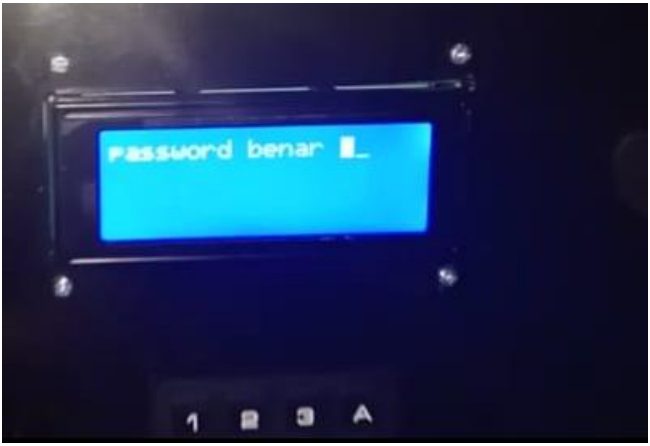
Setelah memasukkan suara langkah selanjutnya adalah memasukkan password setiap pengguna memiliki password yang berbeda, dapat dilihat pada Tabel 4.7 dan gambar memasukkan password dapat dilihat pada Gambar 4.18.

Tabel 4.7 Password Pengguna

No	Pengguna	Kata yang diucapkan	Pasword
1	Gusti	Paring	123A
2	Prayoga	Yoga	456B
3	Bapak Hendra	Hendra	789C

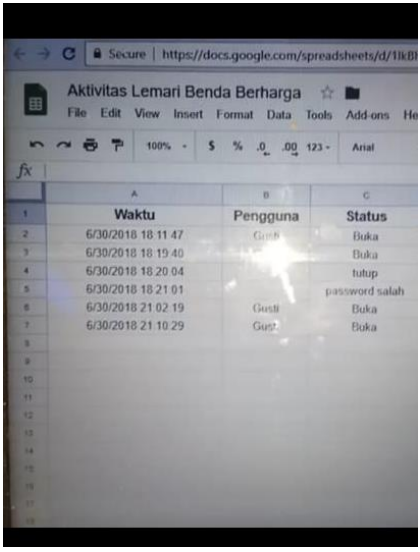
Gambar 4.18 Memasukkan Password

Setelah password benar maka lemari benda berharga akan terbuka dan mengirim data status pada google sheet. Dapat dilihat pada Gambar 4.19, Gambar 4.20, Gambar 4.21



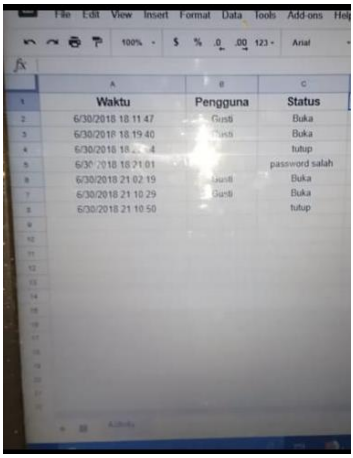
Gambar 4.19 Password Benar

Gambar 4.20 Loker Terbuka



Gambar 4.21 Status Terbuka

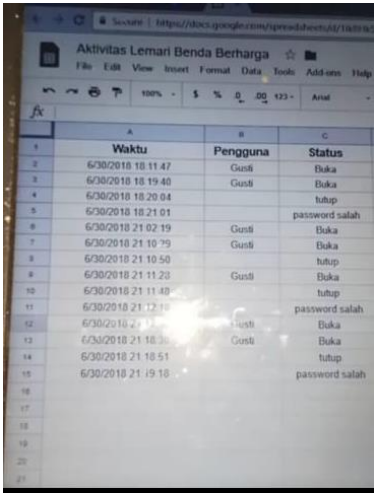
Setelah itu menutup pintu dari lemari benda berharga maka akan mengirim status tutup pada google sheet seperti yang ditunjukkan pada Gambar 4.22.



	A	B	C
1	Waktu	Pengguna	Status
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:20:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup

Gambar 4.22 Status Tutup

Ketika pengguna salah memasukkan password maka *device* akan mengirim pada google sheet yang menunjukkan bahwa password salah yang ditunjukkan pada Gambar 4.23.



Aktivitas Lemari Benda Berharga

	A	B	C
1	Waktu	Pengguna	Status
2	6/30/2018 18:11:47	Gusti	Buka
3	6/30/2018 18:19:40	Gusti	Buka
4	6/30/2018 18:20:04		tutup
5	6/30/2018 18:21:01		password salah
6	6/30/2018 21:02:19	Gusti	Buka
7	6/30/2018 21:10:29	Gusti	Buka
8	6/30/2018 21:10:50		tutup
9	6/30/2018 21:11:23	Gusti	Buka
10	6/30/2018 21:11:48		tutup
11	6/30/2018 21:12:19		password salah
12	6/30/2018 21:18:30	Gusti	Buka
13	6/30/2018 21:18:36	Gusti	Buka
14	6/30/2018 21:18:51		tutup
15	6/30/2018 21:19:18		password salah

Gambar 4.23 Password Salah

4.7 *Usability Testing*

Usability testing berfungsi untuk mengetahui jika lemari benda berharga merupakan alat yang dapat dipasarkan atau termasuk *user friendly*. Pada pengujian ini dilakukan dengan membuat kuisioner yang ditunjukkan pada Tabel 4.8. Pada pengujian ini mengambil sepuluh data koresponden yang nantinya akan diberikan langkah – langkah penggunaan dari lemari benda berharga.

Tabel 4.8 Kuisioner *Usability Testing*

No	Pertanyaan	Baik	Cukup Baik	Kurang Baik	Tidak
1	Apakah desain memenuhi standart keamanan				
2	Apakah sistem sudah berjalan sempurna				
3	Apakah sistem keamanan sudah handal				
4	Apakah pengoprasian sudah usable				
5	Apakah pengenalan suara sudah baik				
6	Apakah pelatihan suara sudah praktis				
7	Bagaimana aktivitas pada google sheet				
8	Bagaimana dengan emergency condition				

Dari sepuluh koresponden menyatakan untuk desain cukup baik yaitu 70%, sistem berjalan sempurna 80% cukup baik, sistem keamanan handal 90% kurang baik, pengoprasian sudah usable 100%, pengenalan suara sudah baik 60% cukup baik, pelatihan suara 80% tidak baik, aktivitas pada google sheet 70% cukup baik dan emergency condition 50% cukup baik.

BAB V
PENUTUP

Dari perancangan Lemari benda berharga dengan sistem keamanan berbasis pengenalan suara menggunakan SMT32F407VG dan modul pengenalan suara V3 didapatkan hasil bahwa modul V3 mengenali suara berdasarkan intonasi dari seseorang dan kerasnya suara yang telah dilatih. Suara yang dapat dikenali yaitu dengan intensitas suara maksimal 88,9 dB dengan tingkat intensitas kebisingan 85,5 dB dengan jarak berbicara kurang dari 5 cm dari mikrofon. Tingkat keberhasilan suara pengguna yang dapat dikenali sebesar 80%. Untuk pengujian sistem dapat berjalan sesuai dengan yang diinginkan. Ketika lemari benda berharga membuka, akan mengirim pencatatan waktu bahwa lemari terbuka oleh diantara tiga pengguna.

Saran untuk penelitian selanjutnya , sebaiknya ditambahkan rangkaian *Band Pass Filter*(BPF) sebelum mikrofon masuk pada modul, untuk menghindari suara yang tidak diinginkan masuk pada saat pelatihan suara pada modul V3. Dan dapat menambahkan fitur google apps untuk lebih mempermudah melihat aktifitas dari lemari benda berharga.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

[1] "Voice Acoustics: an introduction to the science of speech and singing." [Daring]. Tersedia pada: http://www.animations.physics.unsw.edu.au/jw/voice.html. [Diakses: 16-Mar-2018].

[2] SUWADI, *PENGOLAHAN SINYAL DIGITAL*. Institut Teknologi Sepuluh Nopember: Tim Pengampu Mata Kuliah.

[3] N. Jayant, "Delta modulation of pitch, formant, and amplitude signals for the synthesis of voiced speech," *IEEE Trans. Audio Electroacoustics*, vol. 21, no. 3, hlm. 135–140, Jun 1973.

[4] L. D. Paarmann, *Design and analysis of analog filters: a signal processing perspective*. New York: Kluwer Academic Publishers, 2003.

[5] "STM32F4 - ARM Cortex-M4 High-Performance MCUs - STMicroelectronics." [Daring]. Tersedia pada: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series.html?querycriteria=productId=SS1577. [Diakses: 16-Mar-2018].

[6] L. R. Rabiner dan R. W. Schafer, "Introduction to Digital Speech Processing," *Found. Trends® Signal Process.*, vol. 1, no. 1–2, hlm. 1–194, 2007.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1. Program Keseluruhan

```
        * @file           : main.c
        * @brief          : Main program body

******************************************************************
      /* Includes ------------------------------------------------------------
      #include "main.h"
      #include "stm32f4xx_hal.h"
      #include "usart.h"
      #include "gpio.h"
      #include "STM_MY_LCD16X2.h"
      #include "MY_Keypad4x4.h"
      #include <stdbool.h>
      #ifdef __GNUC__
          #define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
      #else
          #define PUTCHAR_PROTOTYPE int fputc( int ch, FILE *f)
      #endif
          PUTCHAR_PROTOTYPE
      {
       HAL_UART_Transmit(&huart3, (uint8_t *)&ch,1,0xFFFF);
          return ch;
      }
          void load();

      /* USER CODE BEGIN Includes */

      /* USER CODE END Includes */

      /* Private variables ---------------------------------------------------------
*/

      /* USER CODE BEGIN PV */
      /* Private variables ---------------------------------------------------------
*/

      /* USER CODE END PV */
```

```
/* Private function prototypes -----------------------------------------
---*/
void SystemClock_Config(void);
void sendA();
void sendB();
void sendC();
void sendD();
void sendE();
void sendF();
/* USER CODE BEGIN PFP */
/* Private function prototypes -----------------------------------------
---*/

/* USER CODE END PFP */
int xx;
int yy;
int zz;
int ww;
int vv;
int kk;
uint8_t rBuf[20];
uint8_t on[20];
uint8_t ind[20];
uint8_t adn[20];
uint8_t off[20];

bool mySwitches[16];
int pass[4];
int pass2[4];
int pass3[4];
int pass4[4];
int pass5[4];
int pass6[4];
int ganti[2];
int ganti1[2];
int ganti2[2];
int ganti3[2];
int ganti4[2];
```

```
int g1;
int g2;
int g3;
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  *
  * @retval None
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
    Keypad_WiresTypeDef myKeypadStruct;
  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

    //data gusti
    on[0]=0xAA;
    on[1]=0x09;
    on[2]=0x0D;
    on[3]=0x00;
    on[4]=0xFF;
    on[5]=0x00;
    on[6]=0x00;
    on[7]=0x02;
    on[8]=0x6F;
    on[9]=0x6E;
    on[10]=0x0A;
```

```
//index yoga
off[0]=0xAA;
off[1]=0x09;
off[2]=0x0D;
off[3]=0x00;
off[4]=0xFF;
off[5]=0x01;
off[6]=0x01;
off[7]=0x02;
off[8]=0x6F;
off[9]=0x6E;
off[10]=0x0A;

//data andre
adn[0]=0xAA;
adn[1]=0x09;
adn[2]=0x0D;
adn[3]=0x00;
adn[4]=0xFF;
adn[5]=0x02;
adn[6]=0x02;
adn[7]=0x02;
adn[8]=0x6F;
adn[9]=0x6E;
adn[10]=0x0A;

//data pak hendra
ind[0]=0xAA;
ind[1]=0x09;
ind[2]=0x0D;
ind[3]=0x00;
ind[4]=0xFF;
ind[5]=0x03;
ind[6]=0x03;
ind[7]=0x02;
ind[8]=0x6F;
ind[9]=0x6E;
ind[10]=0x0A;
```

```
  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART2_UART_Init();
  MX_USART3_UART_Init();
  /* USER CODE BEGIN 2 */
    myKeypadStruct.IN0_Port = GPIOD;
    myKeypadStruct.IN1_Port = GPIOD;
    myKeypadStruct.IN2_Port = GPIOD;
    myKeypadStruct.IN3_Port = GPIOD;
    myKeypadStruct.OUT0_Port = GPIOD;
    myKeypadStruct.OUT1_Port = GPIOD;
    myKeypadStruct.OUT2_Port = GPIOD;
    myKeypadStruct.OUT3_Port = GPIOD;

    myKeypadStruct.IN0pin = GPIO_PIN_0;
    myKeypadStruct.IN1pin = GPIO_PIN_1;
    myKeypadStruct.IN2pin = GPIO_PIN_2;
    myKeypadStruct.IN3pin = GPIO_PIN_3;
    myKeypadStruct.OUT0pin = GPIO_PIN_4;
    myKeypadStruct.OUT1pin = GPIO_PIN_5;
    myKeypadStruct.OUT2pin = GPIO_PIN_6;
    myKeypadStruct.OUT3pin = GPIO_PIN_7;

    Keypad4x4_Init(&myKeypadStruct);

    LCD1602_Begin4BIT(GPIOE, GPIO_PIN_0, GPIO_PIN_1,
GPIOB, GPIO_PIN_4, GPIO_PIN_5, GPIO_PIN_6, GPIO_PIN_7);
    load();
  /* USER CODE END 2 */
```

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
xx=0;
   ww=0;
   ganti[0]=0;
   ganti1[0]=3;
   ganti2[0]=7;
   ganti3[0]=11;
   ganti4[0]=15;
   pass2[0]=0;
pass2[1]=1;
pass2[2]=2;
pass2[3]=3;
pass3[0]=4;
pass3[1]=5;
pass3[2]=6;
pass3[3]=7;
pass4[0]=8;
pass4[1]=9;
pass4[2]=10;
pass4[3]=11;
   pass6[0]=12;
   pass6[1]=13;
   pass6[2]=14;
   pass6[3]=15;
   g1 = 0;
   g2 = 0;
   g3 = 0;
while (1)
{   xx=0;
                        LCD1602_setCursor(1,1);
                        LCD1602_print("masukkan suara : ");

   Keypad4x4_ReadKeypad(mySwitches);
                                for(uint8_t j=0; j<16; j++)
                                {
                                        if(mySwitches[j])
                                        {
```

```
LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                        ganti[xx]=j;
   xx=xx+1;
                                        HAL_Delay(100);

                                        }
                                }

 if (ganti[0]==ganti1[0])
 {xx=0;
            LCD1602_clear();
            LCD1602_print("kode lama 1: ");
            HAL_Delay(2000);
            while(1){
            Keypad4x4_ReadKeypad(mySwitches);
                                for(uint8_t j=0; j<16; j++)
                                {
                                        if(mySwitches[j])
                                        {

LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                        pass5[xx]=j;
   xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {ganti[0]=0;
                                        xx=0;
                                        LCD1602_clear();
                                        break;
                                }
            }
```

```
                    for(uint8_t i=0; i<4; i++)
        {
                if(pass5[i]==pass2[i])
                {
                        ww=ww+1;
                }

        }
        if (ww>3){
         xx=0;
         HAL_Delay(2000);
         LCD1602_clear();
         LCD1602_print("kode baru 1 : ");
          while (1)
         {      Keypad4x4_ReadKeypad(mySwitches);
                        for(uint8_t j=0; j<16; j++)
                        {
                                if(mySwitches[j])
                                {

LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                pass2[xx]=j;
   xx=xx+1;
                                HAL_Delay(1000);

                                }
                        }
                        if (xx==4)
                        {ganti[0]=0;
                                xx=0;
                                LCD1602_clear();
                                break;
                        }}}
        else {
                LCD1602_clear();
         LCD1602_print("kode salah!!! ");
                HAL_Delay(1000);
```

```
                xx=0;
        }

ww=0;
}
        else if (ganti[0]==ganti2[0])
{ xx=0;
        LCD1602_clear();
        LCD1602_print("kode lama 2: ");
        HAL_Delay(2000);
        while(1){
        Keypad4x4_ReadKeypad(mySwitches);
                        for(uint8_t j=0; j<16; j++)
                        {
                                if(mySwitches[j])
                                {

LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                pass5[xx]=j;
   xx=xx+1;
                                HAL_Delay(1000);

                                }
                        }
                        if (xx==4)
                        {ganti[0]=0;
                                xx=0;
                                LCD1602_clear();
                                break;
                        }
        }
                        for(uint8_t i=0; i<4; i++)
        {
                if(pass5[i]==pass3[i])
                {
                        ww=ww+1;
                }
```

```
        }
        if (ww>3){
         xx=0;
         HAL_Delay(2000);
         LCD1602_clear();
         LCD1602_print("kode baru2 : ");
         while (1)
         {        Keypad4x4_ReadKeypad(mySwitches);
                        for(uint8_t j=0; j<16; j++)
                        {
                                if(mySwitches[j])
                                {

LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                pass3[xx]=j;
   xx=xx+1;
                                HAL_Delay(1000);

                                }
                        }
                        if (xx==4)
                        {ganti[0]=0;
                                xx=0;
                                LCD1602_clear();
                                break;
                        }}}
        else {
                LCD1602_clear();
         LCD1602_print("kode salah!!! ");
                HAL_Delay(1000);
                xx=0;}
        ww=0;
 }
        else if (ganti[0]==ganti3[0])
 { xx=0;
         LCD1602_clear();
```

```
            LCD1602_print("kode lama 3: ");
            HAL_Delay(2000);
            while(1){
            Keypad4x4_ReadKeypad(mySwitches);
                                    for(uint8_t j=0; j<16; j++)
                                    {
                                            if(mySwitches[j])
                                            {

LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                            pass5[xx]=j;
   xx=xx+1;
                                            HAL_Delay(1000);

                                            }
                                    }
                                    if (xx==4)
                                    {ganti[0]=0;
                                            xx=0;
                                            LCD1602_clear();
                                            break;
                                    }
             }
                                    for(uint8_t i=0; i<4; i++)
            {
                        if(pass5[i]==pass4[i])
                        {
                                    ww=ww+1;
                        }

            }
            if (ww>3){
             xx=0;
             HAL_Delay(2000);
             LCD1602_clear();
             LCD1602_print("masukkan kode3 : ");
             while (1)
```

```
        {       Keypad4x4_ReadKeypad(mySwitches);
                        for(uint8_t j=0; j<16; j++)
                        {
                                if(mySwitches[j])
                                {

LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                pass4[xx]=j;
   xx=xx+1;
                                HAL_Delay(1000);

                                }
                        }
                        if (xx==4)
                        {ganti[0]=0;
                                xx=0;
                                LCD1602_clear();
                                break;
                        }}}
                        else {
                LCD1602_clear();
          LCD1602_print("kode salah!!! ");
                HAL_Delay(1000);
                xx=0;}
          ww=0;
          }
          else if (ganti[0]==ganti4[0])
{ xx=0;
          LCD1602_clear();
          LCD1602_print("kode lama 4: ");
          HAL_Delay(2000);
          while(1){
          Keypad4x4_ReadKeypad(mySwitches);
                        for(uint8_t j=0; j<16; j++)
                        {
                                if(mySwitches[j])
                                {
```

```
LCD1602_setCursor(2,(xx+1));

LCD1602_print(Keypad4x4_GetChar(j));
                                        pass5[xx]=j;
   xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {ganti[0]=0;
                                        xx=0;
                                        LCD1602_clear();
                                        break;
                                }
          }
                                for(uint8_t i=0; i<4; i++)
          {
                        if(pass5[i]==pass6[i])
                        {
                                ww=ww+1;
                        }

          }
          if (ww>3){
           xx=0;
           HAL_Delay(2000);
           LCD1602_clear();
           LCD1602_print("masukkan kode3 : ");
           while (1)
           {        Keypad4x4_ReadKeypad(mySwitches);
                                for(uint8_t j=0; j<16; j++)
                                {
                                        if(mySwitches[j])
                                        {

LCD1602_setCursor(2,(xx+1));
```

```
   LCD1602_print(Keypad4x4_GetChar(j));
                                        pass6[xx]=j;
      xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {ganti[0]=0;
                                        xx=0;
                                        LCD1602_clear();
                                        break;
                                }}}
                                else {
                        LCD1602_clear();
                 LCD1602_print("kode salah!!! ");
                        HAL_Delay(1000);
                        xx=0;}
                ww=0;
                }


   zz=0;
   yy=0;
 vv=0;
   kk=0;
   pass[0]=0;
   pass[1]=0;
   pass[2]=0;
   pass[3]=0;
   rBuf[0]=0x00;
   rBuf[1]=0x00;
   rBuf[2]=0x00;
   rBuf[3]=0x00;
   rBuf[4]=0x00;
   rBuf[5]=0x05;
   rBuf[6]=0x05;
   rBuf[7]=0x00;
```

```
rBuf[8]=0x00;
rBuf[9]=0x00;
rBuf[10]=0x00;
rBuf[11]=0x00;
rBuf[12]=0x00;
rBuf[13]=0x00;

HAL_UART_Receive(&huart2, rBuf, sizeof(rBuf), 1000);
for (int i = 5;i<7;i++)
{
 if(rBuf[i]==on[i]){
                        zz=zz+1;
                                }
else if(rBuf[i]==off[i]){
            yy=yy+1;
                        }
            else if(rBuf[i]==ind[i]){
            vv=vv+1;
            }
            else if(rBuf[i]==adn[i]){
            kk=kk+1;
                        }
            }

if (yy>1){
                        ww=0;
              xx=0;
                        LCD1602_clear();
                        LCD1602_setCursor(1,1);
                        LCD1602_print("masukkan pass2 : ");
                                    while(1)
                                    {

Keypad4x4_ReadKeypad(mySwitches);
                                    for(uint8_t j=0; j<16; j++)
                                    {
                                                if(mySwitches[j])
                                                {
```

```
        LCD1602_setCursor(2,(xx+1));

        LCD1602_print(Keypad4x4_GetChar(j));
                                        pass[xx]=j;

                                        xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {
                                        break;
                                        }
                                }
                for(uint8_t i=0; i<4; i++)
                {
                        if(pass[i]==pass3[i])
                        {
                                ww=ww+1;
                        }

                }
                if (ww>3)
                {
        //
        HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_SET
);
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password benar ");
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                        sendB();
                        HAL_Delay(1000);
                        while (HAL_GPIO_ReadPin(GPIOD,
GPIO_PIN_15) == GPIO_PIN_RESET){
```

```
                                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                                }
                                HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_RESET);
                                sendD();
                                LCD1602_clear();
                                ww=0;
                                HAL_Delay(1000);
                        }
                                else if(ww<=3)
                                {
                                LCD1602_clear();

                                LCD1602_setCursor(1,1);
                                LCD1602_print("password salah ");
                                sendE();
                                        ww=0;
                                        HAL_Delay(2000);
                                LCD1602_clear();
                                }
        }
                else if (vv>1){
                                ww=0;
                                xx=0;
                                LCD1602_clear();
                                LCD1602_setCursor(1,1);
                                LCD1602_print("masukkan pass3 : ");
                                        while(1)
                                        {

        Keypad4x4_ReadKeypad(mySwitches);
                                        for(uint8_t j=0; j<16; j++)
                                        {
                                                if(mySwitches[j])
                                                {

        LCD1602_setCursor(2,(xx+1));
```

```
        LCD1602_print(Keypad4x4_GetChar(j));
                                        pass[xx]=j;

                                        xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {
                                        break;
                                        }
                                }
                for(uint8_t i=0; i<4; i++)
                {
                        if(pass[i]==pass4[i])
                        {
                                ww=ww+1;
                        }

                }
                if (ww>3)
                {

        //HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_SE
T);
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password benar ");
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                        sendC();
                        HAL_Delay(1000);
                        while           (HAL_GPIO_ReadPin(GPIOD,
GPIO_PIN_15) == GPIO_PIN_RESET){
                                HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
```

```
                        }
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_RESET);
                        sendD();
                                LCD1602_clear();
    ww=0;
                        HAL_Delay(1000);
                }
                        else if(ww<=3)
                        {
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password salah ");
                                sendE();
                                ww=0;
                                HAL_Delay(2000);
                                        LCD1602_clear();

                        }
        }
                else if (zz>1){
                        ww=0;
                  xx=0;
                        LCD1602_clear();
                        LCD1602_setCursor(1,1);
                        LCD1602_print("masukkan pass1 : ");
                                while(1)
                                {

        Keypad4x4_ReadKeypad(mySwitches);
                                for(uint8_t j=0; j<16; j++)
                                {
                                        if(mySwitches[j])
                                        {

        LCD1602_setCursor(2,(xx+1));

        LCD1602_print(Keypad4x4_GetChar(j));
```

```
                                        pass[xx]=j;

                                        xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {
                                        break;
                                        }
                                }
                for(uint8_t i=0; i<4; i++)
                {
                        if(pass[i]==pass2[i])
                        {
                                ww=ww+1;
                        }

                }
                 if (ww>3)
                {

        //HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_SE
T);
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password benar ");
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                        sendA();
                        HAL_Delay(1000);
                        while           (HAL_GPIO_ReadPin(GPIOD,
GPIO_PIN_15) == GPIO_PIN_RESET){
                                HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                        }
```

```
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_RESET);
                        sendD();
                        LCD1602_clear();
                        ww=0;
                        HAL_Delay(1000);
                }
                        else if(ww<=3)
                        {
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password salah ");
                                sendE();
                                ww=0;
                                HAL_Delay(2000);
                                        LCD1602_clear();
                                                }

        else if (kk>1){
                        ww=0;
                  xx=0;
                                LCD1602_clear();
                                LCD1602_setCursor(1,1);
                                LCD1602_print("masukkan pass4 : ");
                                        while(1)
                                        {

            Keypad4x4_ReadKeypad(mySwitches);
                                        for(uint8_t j=0; j<16; j++)
                                        {
                                                if(mySwitches[j])
                                                {

            LCD1602_setCursor(2,(xx+1));

            LCD1602_print(Keypad4x4_GetChar(j));
                                                pass[xx]=j;
```

```
                                        xx=xx+1;
                                        HAL_Delay(1000);

                                        }
                                }
                                if (xx==4)
                                {
                                        break;
                                        }
                                }
                for(uint8_t i=0; i<4; i++)
                {
                        if(pass[i]==pass6[i])
                        {
                                ww=ww+1;
                        }

                }
                 if (ww>3)
                {

        //HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13,GPIO_PIN_SE
T);
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password benar ");
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                        HAL_Delay(1000);
                        sendF();
                        while           (HAL_GPIO_ReadPin(GPIOD,
GPIO_PIN_15) == GPIO_PIN_RESET){
                                HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_SET);
                        }
                        HAL_GPIO_WritePin(GPIOD,
GPIO_PIN_13, GPIO_PIN_RESET);
                        sendD();
```

```
                        LCD1602_clear();
                        ww=0;
                        HAL_Delay(1000);
                }
                        else if(ww<=3)
                        {
                        LCD1602_clear();

                        LCD1602_setCursor(1,1);
                        LCD1602_print("password salah ");
                        sendE();
                                ww=0;
                                HAL_Delay(2000);
                                        LCD1602_clear();
                                                }

    }

  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
}
  }
  /* USER CODE END 3 */

}

void load ()
{
    //clear recognizer//
    uint8_t pData[1];
    pData[0]=0xAA;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x02;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x31;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
```

```
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(1000);

//load index 0//
pData[0]=0xAA;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x03;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x30;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x00;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(1000);

//load index 1//
pData[0]=0xAA;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x03;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x30;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x01;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(10);
pData[0]=0x0A;
HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
HAL_Delay(1000);
```

```
    //load index 2//
    pData[0]=0xAA;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x03;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x30;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x02;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x0A;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
            HAL_Delay(1000);

            //load index 3//
    pData[0]=0xAA;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x03;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x30;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x03;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
    HAL_Delay(10);
    pData[0]=0x0A;
    HAL_UART_Transmit(&huart2,pData, sizeof(pData),10);
            HAL_Delay(1000);


}


void sendA()
```

```
    {

        printf("AT+CIPMODE=0\r\n");
        HAL_Delay(1000);
        printf("AT+CIPMUX=1\r\n");
        HAL_Delay(1000);
        printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\
r\n");
        HAL_Delay(4000);
        printf("AT+CIPSEND=4,131\r\n");
        HAL_Delay(2000);
        //print A
        printf("GET
/pushingbox?devid=vFF6E7B22AE7CAAC&id=1IkBHk5A4bYIZFrvjZ
IxwrcZWBcX3U2zSNBwR3kA-e_0&key=A          HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
    }

    void sendB()
    {
        printf("AT+CIPMODE=0\r\n");
        HAL_Delay(1000);
        printf("AT+CIPMUX=1\r\n");
        HAL_Delay(1000);
        printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\
r\n");
        HAL_Delay(4000);
        printf("AT+CIPSEND=4,131\r\n");
        HAL_Delay(2000);
        //print B
        printf("GET
/pushingbox?devid=v228B5E40CF7CBE1&id=1IkBHk5A4bYIZFrvjZI
xwrcZWBcX3U2zSNBwR3kA-e_0&key=A          HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
    }

    void sendC()
    {
        printf("AT+CIPMODE=0\r\n");
```

```
            HAL_Delay(1000);
            printf("AT+CIPMUX=1\r\n");
            HAL_Delay(1000);
            printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\
r\n");
            HAL_Delay(4000);
            printf("AT+CIPSEND=4,131\r\n");
            HAL_Delay(2000);
            //print C
            printf("GET
/pushingbox?devid=vE1DB8545176D2A0&id=1IkBHk5A4bYIZFrvjZI
xwrcZWBcX3U2zSNBwR3kA-e_0&key=A            HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
        }

        void sendD()
        {
            printf("AT+CIPMODE=0\r\n");
            HAL_Delay(1000);
            printf("AT+CIPMUX=1\r\n");
            HAL_Delay(1000);
            printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\
r\n");
            HAL_Delay(4000);
            printf("AT+CIPSEND=4,131\r\n");
            HAL_Delay(2000);
            //print D
            printf("GET
/pushingbox?devid=v4C357AA26277509&id=1IkBHk5A4bYIZFrvjZIx
wrcZWBcX3U2zSNBwR3kA-e_0&key=A            HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
        }

        void sendE()
        {
            printf("AT+CIPMODE=0\r\n");
            HAL_Delay(1000);
            printf("AT+CIPMUX=1\r\n");
            HAL_Delay(1000);
```

```
            printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\
r\n");
            HAL_Delay(4000);
            printf("AT+CIPSEND=4,131\r\n");
            HAL_Delay(2000);
            //print E
            printf("GET
/pushingbox?devid=vD4FD9E7AC89160D&id=1IkBHk5A4bYIZFrvjZ
IxwrcZWBcX3U2zSNBwR3kA-e_0&key=A            HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
        }

        void sendF()
        {
            printf("AT+CIPMODE=0\r\n");
            HAL_Delay(1000);
            printf("AT+CIPMUX=1\r\n");
            HAL_Delay(1000);
            printf("AT+CIPSTART=4,\"TCP\",\"api.pushingbox.com\",80\
r\n");
            HAL_Delay(4000);
            printf("AT+CIPSEND=4,131\r\n");
            HAL_Delay(2000);
            //print E
            printf("GET
/pushingbox?devid=vD4FD9E7AC89160D&id=1IkBHk5A4bYIZFrvjZ
IxwrcZWBcX3U2zSNBwR3kA-e_0&key=A            HTTP/1.1\r\nHost:
api.pushingbox.com\r\n\r\n");
        }


        /**
          * @brief System Clock Configuration
          * @retval None
          */
        void SystemClock_Config(void)
        {

          RCC_OscInitTypeDef RCC_OscInitStruct;
```

```
  RCC_ClkInitTypeDef RCC_ClkInitStruct;

   /**Configure the main internal regulator output voltage
   */
  __HAL_RCC_PWR_CLK_ENABLE();


__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_
VOLTAGE_SCALE1);

   /**Initializes the CPU, AHB and APB busses clocks
   */
  RCC_OscInitStruct.OscillatorType                    =
RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = 16;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }

   /**Initializes the CPU, AHB and APB busses clocks
   */
  RCC_ClkInitStruct.ClockType                         =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK

|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource                      =
RCC_SYSCLKSOURCE_HSI;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if                (HAL_RCC_ClockConfig(&RCC_ClkInitStruct,
FLASH_LATENCY_0) != HAL_OK)
  {
    _Error_Handler(__FILE__, __LINE__);
  }
```

```
    /**Configure the Systick interrupt time
    */
   HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

    /**Configure the Systick
    */

HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK
);

    /* SysTick_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
  }

  /* USER CODE BEGIN 4 */

  /* USER CODE END 4 */

  /**
    * @brief  This function is executed in case of error occurrence.
    * @param  file: The file name as string.
    * @param  line: The line in file as a number.
    * @retval None
    */
  void _Error_Handler(char *file, int line)
  {
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error
return state */
    while(1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
  }

  #ifdef  USE_FULL_ASSERT
  /**
```

```
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t* file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/**
  * @}
  */

/**
  * @}
  */

/************************************END OF FILE****/
```

A.2. Datasheet ESP8266 01

1.2. Parameters

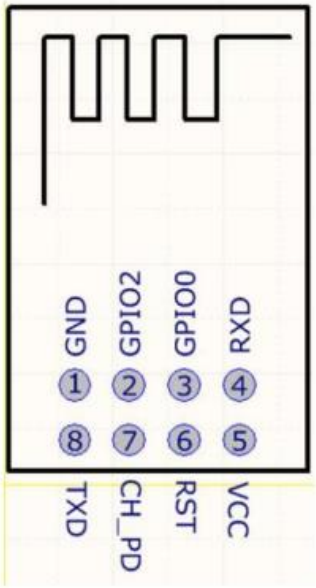
Table 1 below describes the major parameters.

Table 1 Parameters

Categories	Items	Values
WiFi Paramters	WiFi Protocles	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)
Hardware Paramaters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Contorl
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	14.3mm*24.8mm*3mm
	External Interface	N/A
Software Parameters	Wi-Fi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network) / download and write firmware via host
	Ssoftware Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

2. Pin Descriptions

There are altogether 8 pin counts, the definitions of which are described in Table 2 below.



Set	Inquiry	Test	Execute
AT+<x>=<...>	AT+<x>?	AT+<x>=?	AT+<x>
AT+CWMODE=<mode>	AT+CWMODE?	AT+CWMODE=?	-
Set the network mode	Check current mode	Return which modes supported	-

Commands

carefully there are must be no any spaces between the " and IP address or port

Commands	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
AT+RST	restart the module	basic	-	-	-	-	
AT+CWMODE	wifi mode	wifi	AT+CWMODE=<mode>	AT+CWMODE?	AT+CWMODE=?	1= Sta, 2= AP, 3=both	
AT+CWJAP	join the AP	wifi	AT+ CWJAP =<ssid>,<pwd >	AT+ CWJAP?	-	ssid = ssid, pwd = wifi	

Commands	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
						password	
AT+CWLAP	list the AP	wifi	AT+CWLAP				
AT+CWQAP	quit the AP	wifi	AT+CWQAP	-	AT+CWQAP=?		
AT+CWSAP	set the parameters of AP	wifi	AT+ CWSAP=<ssid>,<pwd>, <chl>, <ecn>	AT+ CWSAP?		ssid, pwd, chl = channel, ecn = encryption	Connect to your router: : AT+CWJAP="YOURSSID", "helloworld"; and check if connected: AT+CWJAP?
AT+CIPSTATUS	get the connection status	TCP/IP	AT+ CIPSTATUS				
AT+CIPSTART	set up TCP or UDP connection	TCP/IP	1)single connection (+CIPMUX=0) AT+CIPSTART=<type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART=<id><type>,<addr>, <port>	-	AT+CIPSTART=?	id = 0-4, type = TCP/UDP, addr = IP address, port= port	Connect to another TCP server, set multiple connection first: AT+CIPMUX=1; connect: AT+CIPSTART=4,"TCP","X1.X2.X3.X4",9999
AT+CIPSEND	send data	TCP/IP	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND=		AT+CIPSEND=?		send data: AT+CIPSEND=4,15 and then enter the data

Commands	Description	Type	Set/Execute	Inquiry	test	Parameters	Examples
			<id>,<length>				
AT+CIPCLOSE	close TCP or UDP connection	TCP/IP	AT+CIPCLOSE=<id> or AT+CIPCLOSE		AT+CIPCLOSE=?		
AT+CIFSR	Get IP address	TCP/IP	AT+CIFSR		AT+CIFSR=?		
AT+CIPMUX	set mutiple connection	TCP/IP	AT+CIPMUX=<mode>	AT+CIPMUX?		0 for single connection 1 for mutiple connection	
AT+CIPSERVER	set as server	TCP/IP	AT+CIPSERVER=<mode>[,<port>]			mode 0 to close server mode, mode 1 to open; port = port	turn on as a TCP server: AT+CIPSERVER=1,8888, check the self server IP address: AT+CIFSR=?
+IPD	received data						

A.3. Datasheet STM32F407VG

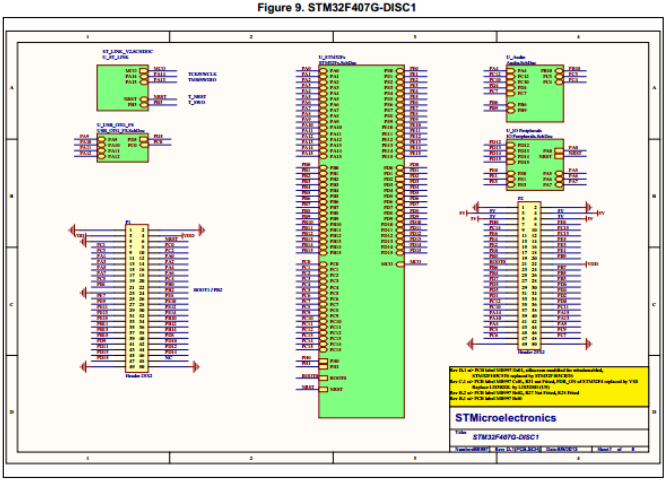
1 Features

The STM32F4DISCOVERY offers the following features:

- STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex® -M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package
- On-board ST-LINK/V2 on STM32F4DISCOVERY or ST-LINK/V2-A on STM32F407G-DISC1
- ARM® mbed Enabled™ (http://mbed.org) with ST-LINK/V2-A only
- USB ST-LINK with re-enumeration capability and three different interfaces:
  - Virtual COM port (with ST-LINK/V2-A only)
  - Mass storage (with ST-LINK/V2-A only)
  - Debug port
- Board power supply:
  - Through USB bus
  - External power sources:
    3 V and 5 V
- LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
- MP45DT02 ST MEMS audio sensor omni-directional digital microphone
- CS43L22 audio DAC with integrated class D speaker driver
- Eight LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power on
  - Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  - 2 USB OTG LEDs LD7 (green) VBUS and LD8 (red) over-current
- Two push buttons (user and reset)
- USB OTG FS with micro-AB connector
- Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing
- Comprehensive free software including a variety of examples, part of the STM32CubeF4 package or STSW-STM32068 for legacy standard library usage

7 Electrical schematics

Figure 9. STM32F407G-DISC1



A.4. Datasheet Modul Pengenalan Suara V3

Voice Recognition Module V3

Speak to Control (Arduino compatible)

Overview

ELECHOUSE Voice Recognition Module is a compact and easy-control speaking recognition board.

This product is a speaker-dependent voice recognition module. It supports up to 80 voice commands in all. Max 7 voice commands could work at the same time. Any sound could be trained as command. Users need to train the module first before let it recognizing any voice command.

This board has 2 controlling ways: Serial Port (full function), General Input Pins (part of function). General Output Pins on the board could generate several kinds of waves while corresponding voice command was recognized.

Return

| Head (AA) | Length| Command | Data | End (0A) |
Length = L(Length + Command + Data)

NOTE: Data area is different with different with commands.

Code

ALL CODE ARE IN HEXADECIMAL FORMAT

FRAME CODE
AA --> Frame Head
0A --> Frame End

CHECK
00 --> Check System Settings
01 --> Check Recognizer
02 --> Check Record Train Status
03 --> Check Signature of One Record

SYSTEM SETTINGS
10 --> Restore System Settings
11 --> Set Baud Rate
12 --> Set Output IO Mode
13 --> Set Output IO Pulse Width
14 --> Reset Output IO
15 --> Set Power On Auto Load

RECORD OPERATION
20 --> Train One Record or Records
21 --> Train One Record and Set Signature
22 --> Set Signature for Record

RECOGNIZER CONTROL
30 --> Load a Record or Records to Recognizer
31 --> Clear Recognizer
32 --> Group Control

Train One Record and Set Signature (21)

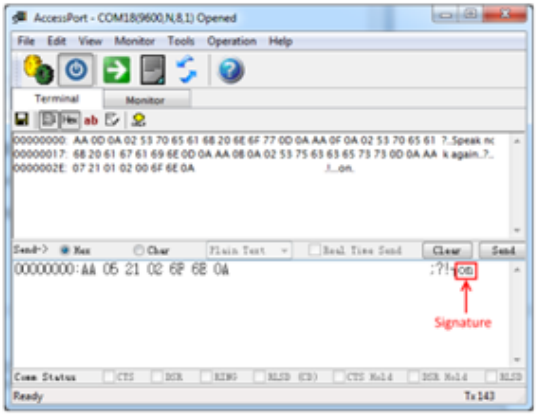
Train one record and set a signature for it, one record one time.
Format:
| AA| 03+SIGLEN | 21 | RECORD | SIG | 0A | (Set signature)
Return:

| AA| LEN | 0A | RECORD | PROMPT | 0A | (train prompt)
| AA| 05+SIGLEN | 21 | N | RECORD | STA | SIG | 0A |

	Description
RECORD	Voice command record index
SIG	Signature string
PROMPT	Prompt string: • Speak now • Speak again • Success
N	Number of successful training voice commands

Example:

Train command 02 with signature "on"



-----Halaman ini sengaja dikosongkan-----

PROFIL PENULIS

Gusti Paring, dilahirkan di Sidoarjo pada 20 Mei 1995. Merupakan anak ketiga dari tiga bersaudara. Gusti pernah menempuh pendidikan di SD Muhammadiyah 1-2 Taman (2001-2007), SMP Negeri 1 Taman (2007-2010), SMA Negeri 1 Taman (2013-2016), setelah lulus SMA Gusti melanjutkan pendidikan di Institut Teknologi Sepuluh Nopember (ITS) Surabaya bidang studi D3 Teknik Elektro Komputer Kontrol. Kemudian melanjutkan program strata-1 (S1) di ITS dan mengambil jurusan Teknik Elektro pada bidang studi Elektronika. Nomor Telepon :082141804942. Email : gustiparing@gmail.com

-----Halaman ini sengaja dikosongkan-----